



AFRL-AFOSR-VA-TR-2016-0109

---

## BRI-FY13-Parallel Multilevel Decomposition Methods for Fluid Plasma Models

**Steve McCormick**  
**FRONT RANGE SCIENTIFIC COMPUTATION INC**

---

**02/26/2016**  
**Final Report**

DISTRIBUTION A: Distribution approved for public release.

Air Force Research Laboratory  
AF Office Of Scientific Research (AFOSR)/ RTA2  
Arlington, Virginia 22203  
Air Force Materiel Command

**REPORT DOCUMENTATION PAGE**
*Form Approved  
OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to the Department of Defense, Executive Service Directorate (0704-0188). Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.**

|   |  |  |   |                            |  |  |
|---|--|--|---|----------------------------|--|--|
| <b>1. REPORT DATE (DD-MM-YYYY)</b><br>01-02-2016  |  |  | <b>2. REPORT TYPE</b><br>Final              |                            | <b>3. DATES COVERED (From - To)</b><br>29-09-2012 to 01-02-2016  |  |
| <b>4. TITLE AND SUBTITLE</b><br>Parallel Multilevel Decomposition Methods for Fluid Plasma Models   |  |  | <b>5a. CONTRACT NUMBER</b>                  |                            |  |  |
|   |  |  | <b>5b. GRANT NUMBER</b><br>FA9550-12-1-0478 |                            |  |  |
|   |  |  | <b>5c. PROGRAM ELEMENT NUMBER</b>           |                            |  |  |
| <b>6. AUTHOR(S)</b><br>McCormick, Stephen, F.<br>Beckwith, Kristian R. C.<br>Olson, Luke, N.  |  |  | <b>5d. PROJECT NUMBER</b>                   |                            |  |  |
|   |  |  | <b>5e. TASK NUMBER</b>                      |                            |  |  |
|   |  |  | <b>5f. WORK UNIT NUMBER</b>                 |                            |  |  |
| <b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b><br>Front Range Scientific Computations, Inc.<br>P. O. Box 503<br>Lake City, CO 81235-0503   |  |  |   |                            | <b>8. PERFORMING ORGANIZATION REPORT NUMBER</b><br>AFFR-001  |  |
| <b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b><br>Air Force Office of Scientific Research<br>875 N. Randolph St. Room 3112<br>Arlington, VA 22203   |  |  |   |                            | <b>10. SPONSOR/MONITOR'S ACRONYM(S)</b><br>AFOSR   |  |
|   |  |  |   |                            | <b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>  |  |
| <b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b><br>Approved for public release; distribution is unlimited DISTIRBUTION A   |  |  |   |                            |  |  |
| <b>13. SUPPLEMENTARY NOTES</b>  |  |  |   |                            |  |  |
| <b>14. ABSTRACT</b><br>1. Multilevel Decomposition. Full implementations of both domain and range versions were developed and the comparative advantages and disadvantages were studied and documented. Numerical tests and model analyses showed these methods to be effective for current and foreseeable large-scale parallel architectures. 2. Adaptive AMG. New adaptive versions of AMG/SA were developed and tested, showing that the large-scale performance characteristics of these full-system methods to be superior to conventional approaches. 3. Jacobian-Free Newton-Krylov Methods. New physics-based preconditioners for the two-fluid problems were developed that exploit the new, ultra-scalable multilevel algorithms. New strategies for reducing preconditioner setup costs during the JFNK iterations were developed and implemented. 4. Space-Time Parallelization. A new multilevel algorithm was developed for implicit time-stepping schemes that coarsens in space and time and its efficacy for space-time grids was demonstrated. 5. Fault Resilience. A software layer for fault detection and recovery was developed that seamlessly integrates into the multilevel decomposition methods. Efficiency of the proposed fault detection and recovery method was carefully demonstrated. |  |  |   |                            |  |  |
| <b>15. SUBJECT TERMS</b>  |  |  |   |                            |  |  |
| <b>16. SECURITY CLASSIFICATION OF:</b><br>a. REPORT    b. ABSTRACT    c. THIS PAGE<br>U                U                U   |  |  | <b>17. LIMITATION OF ABSTRACT</b><br>UU     | <b>18. NUMBER OF PAGES</b> | <b>19a. NAME OF RESPONSIBLE PERSON</b><br>Stephen F. McCormick<br><b>19b. TELEPHONE NUMBER (Include area code)</b><br>(303) 442-0724 |  |

**FINAL REPORT**  
**MULTILEVEL DECOMPOSITION METHODS**  
**FOR FLUID PLASMA MODELS\***

S. MCCORMICK, K. BECKWITH, L. OLSON

**1. Multilevel Decomposition.** Multigrid methods are often well-suited for large-scale scientific computing problems because they offer the possibility of solving equations at a cost that is optimal in the sense that is in small proportion to the number of unknowns. However, the challenge for multigrid and other matrix equation solvers on large parallel machines is that performance can suffer from the high cost of communication relative to that of computation. While Algebraic MultiGrid (AMG [1, 2]) solvers scale nearly optimally, they too are increasingly affected by relative communication costs as the number of processors increase. Indeed, all known parallel multigrid algorithms experience degrading communication costs for the main computation in each V-Cycle. Previous *geometric* multilevel approaches developed by [3, 4, 5, 6, 7, 8, 9, 10, 11] aimed to control this cost by trading communication for computation using redundant processing on overlapping grids. Inspired by these efforts, the Algebraic MultiGrid Domain (AMG-DD) and Range Decomposition (AMG-RD) methods developed in this project achieve the same goal by tasking possibly otherwise idle processors to perform redundant computations via a domain decomposition approach. The basic idea is to use subdomains that fully overlap at coarse scales. The departure from the previous geometric approaches is to exploit the benefits of domain decomposition in a purely *algebraic* AMG setting.

AMG-DD and AMG-RD first assume that the setup for an effective AMG implementation has been formed [1, 2]. The two methods then use the *global* AMG hierarchical constructs (coarse grids, coarse-grid operators, and intergrid transfer operators) to create a composite grid for each processor. Each composite grid consists of the original grid in and about the subdomain owned by its associated processor and of grids that are increasingly coarse as they extend away from the processor subdomain to the boundary. These composite grids are formed algebraically and directly from the hierarchical constructs determined in the traditional AMG setup phase. In this way, AMG-DD and AMG-RD can be thought of as globally overlapping domain decomposition methods that have reduced communication per cycle, since they do not require communication on each grid level as standard AMG methods do. Moreover, the new process provides an efficient communication phase between each cycle, in contrast to the expensive all-to-all communication processes of the previous geometric approaches. The composite grids thus provide a means for maintaining effective communication between processors that controls cost and maintains optimal convergence rates.

The main focus of this effort was the development and study of AMG-DD and AMG-RD as possible alternatives to existing AMG approaches for solving large-scale matrix equations on advanced parallel machines, with fluid plasma models as an ultimate target. To this end, some theoretical properties of these methods were established and serial numerical tests of their convergence properties were analyzed over a spectrum of parameters on a model problem. Also studied were some heuristics and a parameter influences based on a performance model, both of which were designed to anticipate the potential of AMG-DD and AMG-RD for use on emerging parallel architectures.

Since AMG-DD and AMG-RD are constructed on top of an existing AMG hierarchy, the cost of the setup, as in all AMG algorithms, needs to be addressed. In a purely serial setting, setup costs for AMG-DD and AMG-RD can be up to about twice that of standard AMG due to the redundant calculation that we use. However, in parallel, owing to the construction of the algorithm from existing operators, the increased setup cost can be reduced essentially to the cost of communicating the components of the operators needed to each processor. This communication can be done using the same pattern as the residual communication and, as such, is bounded in cost by that of performing one extra V-Cycle.

The numerical test that were performed confirmed that, for current parallel architectures, trading of communication for computation that AMG-DD achieves is advantageous for current and anticipated architectures in a very large computational environment. The results showed that these new methods can be especially effective when used in a nested iteration process where the models are first resolved on crude levels to provide good initial data for increasingly finer resolutions.

**2. Adaptive AMG.** A coupled system of elliptic partial differential equations can be treated by iterating in a sequential or simultaneous way on the individual scalar equations, provided the coupling is not too strong. This is the approach Chacón [12, 13, 14] takes in the solver for his MHD simulations, and it provided the setting for the project’s initial development and testing of the performance and scalability of multilevel decomposition solvers. However, even with fairly weak coupling between the individual equations, it would be more effective to apply the multigrid solvers to the full system. The difficulty in treating a coupled system in this way is the need to develop coarsening strategies that properly address the coupling, and this can be especially challenging for the automatic coarsening strategies that form the basis for algebraic multigrid and smoothed aggregation. To enable a full and potentially more effective multigrid solver for the two-fluid system, the project therefore focused on extending the adaptive version of SA (aSA). The central aim here was to develop an efficient algebraic solver that could be applied to plasma fluid systems with *algorithmically* optimal performance in the sense that its cost is essentially proportional to the size of the discrete system. Such an optimal adaptive solver could then be used with the multilevel decomposition approaches to enable simulations at ultra-scales that would otherwise be derailed by the growing solver costs.

The basic idea behind aSA is to aim for a coarsening strategy that guarantees the so-called weak approximation property, which states that the coarse approximation to a given fine-grid error in the Euclidean norm is smaller than some constant times the energy norm of that error (with the constant assumed to be inversely proportional to the norm of the matrix). This property guarantees certain convergence properties of unsmoothed aggregation and, with subsequent smoothing of interpolation, it tends to produce an optimally convergent SA. In any case, this aim is achieved by starting with an error that is certain to be in need of a coarse-grid correction, which is accomplished by simply applying the initial SA scheme to the homogeneous equation beginning with a random initial guess. If slow convergence is observed, then the interpolation operator is adjusted to improve its approximation to the resulting error. This process is repeated until good convergence is achieved.

Initial tests with this new version of aSA have been very encouraging. With just one or, in some cases, two adaptive cycles, convergence factors are achieved that compare to model problem results. The tests suggest that aSA should provide a very effective solver for systems, especially for coupled plasma models.

**3. Jacobian-Free Newton-Krylov Methods for Fluid Plasmas.** Jacobian-Free, Newton-Krylov (JFNK) methods are based on the Newton-Krylov methods, which were first considered in the work of Brown and Saad [15, 16]. The early advances involved the realization that inexact Newton methods were simpler and cheaper to implement and exhibited similar convergence to exact Newton methods. Researchers also explored preconditioning strategies, such as the Schwarz methods [17], within the Newton-Krylov systems. A detailed history and thorough study of the Jacobian-Free, Newton-Krylov methods is presented by Knoll and Keyes in [18]. The JFNK methods have since been applied to a range of problems from power systems [19] to MHD systems [12, 13, 14].

Here, the JFNK method is used to solve the nonlinear system of equations generated by a fully implicit temporal discretization of the constitutive equations, which exhibit either a hyperbolic, parabolic or elliptic structure, depending of the plasma regime. A nonlinear system  $\mathbf{F}(\mathbf{x}) = \mathbf{0}$  must be solved for the new solution,  $\mathbf{x}$ , at each new time step. The solution of the nonlinear system follows Newton's method whereby the Jacobian system of the form

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \Big|_k \delta \mathbf{x}_k = -\mathbf{F}(\mathbf{x}_k), \quad (3.1)$$

where  $\mathbf{x}_{k+1} = \mathbf{x}_k + \delta \mathbf{x}_k$  is the  $k^{th}$  iteration.

The JFNK method employs an inner Krylov method as the linear solver for the Jacobian system and an outer Newton's method for the nonlinear problem. The key to the approach is that the inner Krylov method requires only matrix-vector products for the Jacobian (linear) system. This can be approximated with the approximate Gateaux derivative as

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \Big|_k \mathbf{v} \approx \frac{\mathbf{F}(\mathbf{x}_k + \epsilon \mathbf{v}) - \mathbf{F}(\mathbf{x}_k)}{\epsilon}. \quad (3.2)$$

The error in this approximation is proportional to  $\epsilon$ . The result of this approximation is a matrix-free implementation of the matrix-vector multiply; however, the entire method will not be matrix-free because a preconditioner is needed to have reasonable convergence and, for wide applicability, the preconditioner will be built based in an algebraic way on a matrix. In this project, we have utilized JFNK solvers developed at Sandia National Laboratory as part of the *Trilinos* library [20], specifically the *Nonlinear Object-Oriented Solutions* (NOX) framework [21, 22], which combines the *AztecOO* iterative solvers package [23] with algebraic multigrid preconditioners available within the *MultiLevel* package (ML, [24]). These solvers have been integrated into the Tech-X USIM tool for fluid-plasma-electromagnetic simulations, developed with prior support from AFOSR under grants FA9550-12-C-0039 and FA9550-14-C-0004 and the Department of Energy under grants DE-SC0000833, DE-SC0009585 and DE-AR0000566. This approach has allowed us to rapidly prototype different physics-based preconditioners to develop fully-implicit solvers for plasma physics problems.

**3.1. Preconditioners for Moving Least Squares Problems.** The algorithms for solving fluid plasma equations in the USIM code rely on moving least squares operators for interpolation of fluid quantities on the computational mesh [25]. As a first step in developing preconditioners for fluid-plasma problems, we investigated constructing preconditioners for a prototype Laplace problem based on this class of operators in order to both verify that these operators are amenable to multigrid preconditioning and to demonstrate the overall scalability of the NOX framework integrated into the

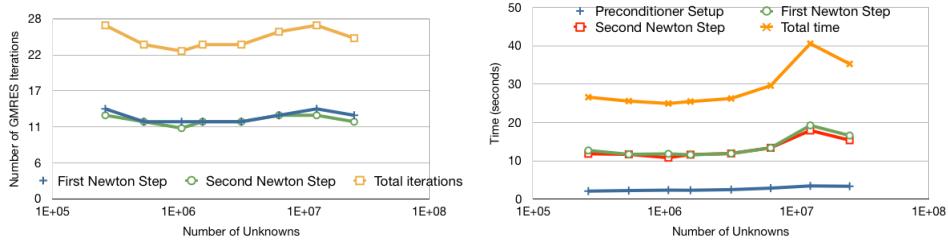


Fig. 3.1: Scalability of ML preconditioned JFNK solver for a moving least squares discretization of a Laplace operator. The left panel shows the number of inner Krylov solves for the first and second outer Newton iteration; the right panel shows the preconditioner setup time, the time taken for the first Newton iteration, the time taken for the second Newton iteration and the total time to solution. Overall, the approach shows good weak scalability.

USIM tool. To introduce the preconditioned moving least squares algorithm, consider two-dimensional data  $Q_i$  on  $N$  points  $(x_i, y_i)$  with weights  $w_i$  where  $i = 0, \dots, N$ . The Vandermonde matrix for this problem takes the form:

$$\begin{bmatrix} w_0 1 & w_0 x_0 & w_0 x_0 y_0 & w_0 x_0^2 & w_0 y_0^2 \\ w_1 1 & w_1 x_1 & w_1 x_1 y_1 & w_1 x_1^2 & w_1 y_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_N 1 & w_N x_N & w_N x_N y_N & w_N x_N^2 & w_N y_N^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_N \end{bmatrix} \quad (3.3)$$

This equation is of the form  $PA = Q$ , with solution  $A = [P^T P]^{-1} P^T Q$ . Let  $B = [P^T P]^{-1} P^T$ , then at a point  $(x, y)$  we can write (using tensor notation):

$$q(x, y) = B_{\alpha\beta} p^\alpha Q^\beta \quad (3.4)$$

$$\frac{\partial^i q(x, y)}{\partial x^i} = B_{\alpha\beta} \frac{\partial^i p^\alpha}{\partial x^i} Q^\beta \quad (3.5)$$

$$p = (1, x, y, xy, x^2, y^2)^T \quad (3.6)$$

In two-dimensions, we can then construct a finite volume discretization of the Laplace operator through:

$$\nabla^2 Q = \frac{1}{dxdy} \int \frac{\partial q(x, y)}{\partial x} \cdot \hat{n} d\ell = \frac{1}{dxdy} \int B_{\alpha\beta} \frac{\partial^i p^\alpha}{\partial x^i} Q^\beta \cdot \hat{n} d\ell \quad (3.7)$$

where  $\hat{n}$  is the outwards facing normal at each cell edge and  $d\ell$  is the length of the edge. In three-dimensions, this operator becomes:

$$\nabla^2 Q = \frac{1}{dV} \int \frac{\partial q(x, y, z)}{\partial x} \cdot \hat{n} dA = \frac{1}{dV} \int B_{\alpha\beta} \frac{\partial^i p^\alpha}{\partial x^i} Q^\beta \cdot \hat{n} dA \quad (3.8)$$

The matrix used to precondition this operator is then constructed from:

$$P_k = \frac{1}{dV} \int B_{\alpha\beta} \frac{\partial^i p^\alpha}{\partial x^i} \cdot \hat{n} dA \quad (3.9)$$

This is then applied within the Jacobian-Free Newton Krylov solver as a right preconditioner:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \Big|_k P_k^{-1} P_k \delta \mathbf{x}_k = -\mathbf{F}(\mathbf{x}_k), \quad (3.10)$$

We construct  $P_k^{-1}$  using Trilinos ML Uncoupled Smoothed Aggregation with 5 levels; Jacobi smoothing is utilized on the coarsest level and Gauss Seidel smoothing on all others. The scalability of this approach for a simple three-dimensional linear Poisson-type problem (e.g.  $\nabla^2 \phi = \rho$  with  $\rho$  held constant) is shown in Figure 3.1. The data of these figures demonstrate that the implementation of the Trilinos infrastructure within the USIM tool exhibits good scalability for three-dimensional problems.

**3.2. Physics-Based Preconditioners for Adiabatic Magnetohydrodynamics.** For density, momentum, and temperature, the resistive MHD equations are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (3.11)$$

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v} + \mathbf{I} p + \Pi) = \mathbf{J} \times \mathbf{B} \quad (3.12)$$

$$\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T + (\gamma - 1) T \nabla \cdot \mathbf{v} = 0. \quad (3.13)$$

The evolution of the magnetic field is described by:

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{v} \times \mathbf{B}; \quad \nabla \times \mathbf{B} = \mu_0 \mathbf{J}; \quad \nabla \cdot \mathbf{B} = 0 \quad (3.14)$$

These equations can be cast in conservative form as  $\partial_t U + \nabla \cdot F(U) = S(U)$ , with:

$$U = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ E \\ \mathbf{B} \end{bmatrix}; \quad F(U) = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho \mathbf{v} + (p + \frac{1}{2} \mathbf{B} \cdot \mathbf{B}) \mathbb{I} - \mathbf{B} \mathbf{B} \\ (E + p + \frac{1}{2} \mathbf{B} \cdot \mathbf{B}) \mathbf{v} - \mathbf{B}(\mathbf{B} \cdot \mathbf{v}) \\ \mathbf{v} \mathbf{B} - \mathbf{B} \mathbf{v} \end{bmatrix}; \quad S(U) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.15)$$

where  $E = \frac{P}{\gamma - 1} + \frac{\rho \mathbf{v} \cdot \mathbf{v}}{2} + \frac{\mathbf{B} \cdot \mathbf{B}}{2}$  is the total energy of the fluid. The constraint,  $\nabla \cdot \mathbf{B} = 0$  requires special mention; physically accurate solution of eqn. 3.11 through 3.14 requires that this constraint is enforced during the evolution. Approaches such as constrained transport [26] have been developed to enforce this constraint to machine precision; however, these methods typically do not extend to implicit time discretization or high order schemes. Instead, we adopt Generalized Lagrange Multiplier schemes due to [27], where the MHD system is augmented to include a scalar potential,  $\psi$  that acts to advect divergence errors out of the grid at a speed  $c_h$  (corresponding to the fastest hyperbolic wave speed in the simulation domain) while simultaneously damping these errors to zero at a rate  $c_h^2/c_p^2$  (where  $c_p^2$  is a problem dependent constant) [28]. With this augmentation, the MHD system becomes:

$$U = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ E \\ \mathbf{B} \\ \psi \end{bmatrix}; \quad F(U) = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho \mathbf{v}^T + (p + \frac{1}{2} \mathbf{B} \cdot \mathbf{B}) \mathbb{I} - \mathbf{B} \mathbf{B}^T \\ (E + p + \frac{1}{2} \mathbf{B} \cdot \mathbf{B}) \mathbf{v} - \mathbf{B}(\mathbf{B} \cdot \mathbf{v}) \\ \mathbf{v} \mathbf{B}^T - \mathbf{B} \mathbf{v}^T + \psi \\ c_h^2 \mathbf{B} \end{bmatrix}; \quad S(U) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -\frac{c_h^2}{c_p^2} \psi \end{bmatrix} \quad (3.16)$$

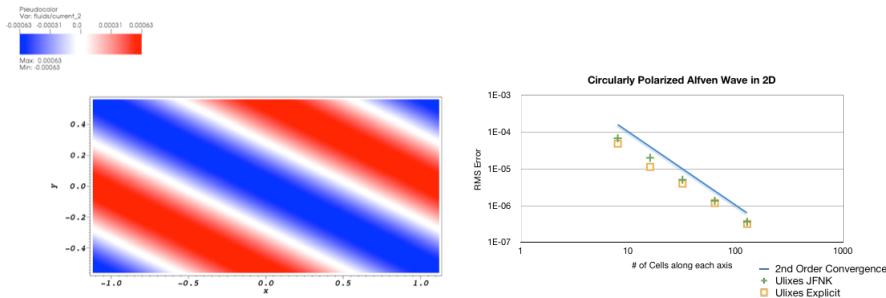


Fig. 3.2: Convergence of a multidimensional circularly polarized Alfvén wave test due to [29]. The left panel shows the current normal to the plane after one complete traversal of the grid. The right panel shows the norm of the L<sub>2</sub>-error as a function of grid resolution, including divergence errors for an explicit second order Runge-Kutta discretization compared to the fully implicit scheme described here. Both discretizations demonstrate second order convergence with comparable errors.

The addition of these terms leads to an augmented system that contains two additional modes, carrying jumps in the normal component of  $\mathbf{B}$  and  $\psi$ . These waves are decoupled into a linear hyperbolic system that can be solved analytically [28]:

$$\partial_t B_x = -\partial_x \psi; \quad \partial_t \psi = -c_h^2 \partial_x B_x \quad (3.17)$$

To solve  $\partial_t U + \nabla \cdot F(U) = S(U)$  using JFNK methods, we cast the system equations as a non-linear functional,  $\mathbb{F}(U) = \partial_t U + \nabla \cdot F(U) - S(U)$ . Adopting a  $\theta$  temporal discretization [12, 13, 14], we have:

$$\begin{aligned} \mathbb{F}(U^{n+1}) &= U^{n+1} - U^n + \theta \Delta t [\nabla \cdot F(U^{n+1}) - S(U^{n+1})] \\ &\quad + (1 - \theta) \Delta t [\nabla \cdot F(U^n) - S(U^n)] \end{aligned} \quad (3.18)$$

where  $\theta = 0, 1$  corresponds to forward/backward Euler respectively, whereas  $\theta = 0.5$  corresponds to Crank-Nicholson. To precondition this non-linear system, we linearize the approximate Gateaux derivative (Eqn. 3.2) acting on the above equation, yielding:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \Big|_k \delta U_k \approx \left\{ 1 - \theta \Delta t \left[ \frac{\partial F(U)}{\partial U} - \frac{\partial S(U)}{\partial U} \right] \right\} \delta U_k \quad (3.19)$$

We again apply this approximate Jacobian through right preconditioning (Eqn. 3.10), requiring the computation of the inverse of Eqn. 3.19. In order to compute the inverse, we note that  $\frac{\partial S(U)}{\partial U}$  has a trivial answer for the system of equations (3.16). Determining the form of  $\frac{\partial F(U)}{\partial U}$  is more complex. USIM makes use of finite volume discretizations and discontinuous Galerkin discretizations as described in [25]. As implemented in USIM, both of these discretization methods make use of Riemann solvers to compute self-similar solutions to the breakdown of a discontinuity. The finite volume scheme (for example) then integrates this solution over a control volume to construct a second (or higher) order accurate approximation to  $\nabla \cdot F(U)$  though (e.g.):

$$\nabla \cdot F(U) = \frac{1}{dV} \int F(U_L, U_R) \cdot \hat{n} dA \quad (3.20)$$

$$F(U_L, U_R) = \frac{1}{2} \left[ F(U_L) + F(U_R) + \left| \tilde{A} \right| (U_L - U_R) \right] \quad (3.21)$$

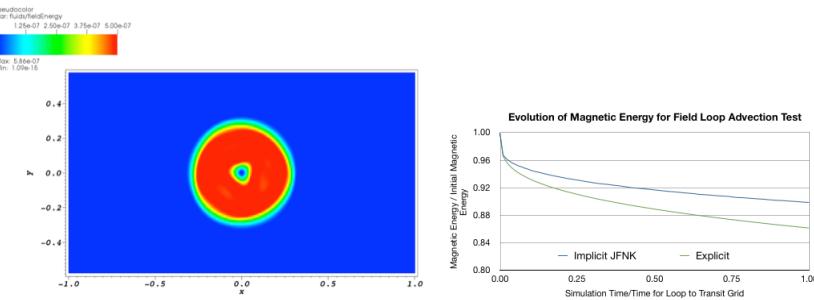


Fig. 3.3: Evolution of a weakly magnetized field loop due to [32, 33]. This test verifies that the divergence of the magnetic field remains zero on the computational stencil; non-zero magnetic field divergence causes rapid growth of magnetic energy and the destruction of the loop. The left panel shows the structure of the magnetic energy on the computational domain after one crossing; the right panel shows the evolution of the magnetic energy integrated over the simulation domain over one crossing period. The data of these two panels demonstrate that the magnetic field loop remains well defined and that the magnetic energy decays due to numerical diffusion, i.e. the divergence of the magnetic field is zero on the computational stencil.

where we have adopted a Roe-type solver [30] to compute a self-similar upwind flux from initial states at the left- ( $U_L$ ) and right- ( $U_R$ ) hand sides of the discontinuity. The reconstruction of the initial data,  $U$ , located at control volume centers to form  $U_L$  and  $U_R$  is performed by the moving least squares algorithm described in §3.1 and coefficients for this data are included in the matrix for forming  $U_L$  and  $U_R$ . Finally, the dissipation matrix,  $|\tilde{A}|$  is formed through  $|\tilde{A}| = \tilde{L} |f(\lambda)| \tilde{R}$ , where  $\tilde{L}, \tilde{R}$  are the left and right eigenvectors of the system and we have applied an entropy fix to the eigenvalues,  $\lambda$  to render the flux differentiable. To construct  $\frac{\partial F(U_L, U_R)}{\partial U}$ , we follow the approach described by [31]:

$$\frac{\partial F(U_L, U_R)}{\partial U_L} = \frac{1}{2} \left[ \frac{\partial F(U_L)}{\partial U_L} + |\tilde{A}| \right]; \quad \frac{\partial F(U_L, U_R)}{\partial U_R} = \frac{1}{2} \left[ \frac{\partial F(U_R)}{\partial U_R} - |\tilde{A}| \right] \quad (3.22)$$

The flux jacobians,  $\frac{\partial F(U_L, U_R)}{\partial U_L}$  and  $\frac{\partial F(U_L, U_R)}{\partial U_R}$  are constructed from the same eigensystem (with the entropy fix applied) as the dissipation matrix,  $|\tilde{A}|$ . The eigensystem for the augmented MHD system (3.16) is well known (e.g. [29, 28]) and can be used directly to construct the matrix associated with Eqn. 3.22.

To precondition the MHD system, we construct  $P_k^{-1}$  using Trilinos ML Uncoupled Smoothed Aggregation with 5 levels; on each level, we use block ILU smoothing with zero overlap and symmetric Gauss-Seidel relaxation on each level and choose the block size to be equal to the number of partial differential equations in the system (here, this is 9). In Fig. 3.2 we demonstrate that our approach exhibits second order accuracy for propagation of a non-linear circular polarized Alfvén wave in multidimensions due to [29]. To our knowledge, this is the first demonstration of a fully implicit algorithm for magnetohydrodynamics with second-order accuracy. In Fig. 3.3 we utilize the field loop advection test due to [32, 33] to demonstrate that that our algorithm preserves the divergence of the magnetic field in multi-dimensions. In Figure 3.4 we demonstrate

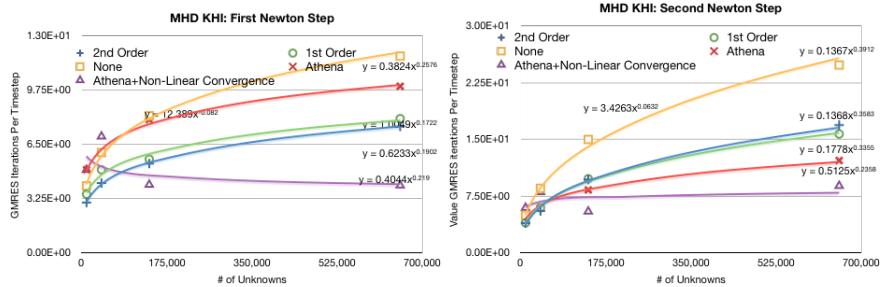


Fig. 3.4: Scalability of a range of different preconditioning strategies for JFNK based solution of the compressible MHD equations used to compute the linear growth stage of the magnetized Kelvin-Helmholtz instability as a function of problem size. The simulation timestep chosen so that highest resolution requires 2 outer Newton iterations. The left and right panel shows the scaling of the number of inner Krylov iterations with problem size for the first and second outer Newton iterations respectively. When combined with adaptive linear convergence criteria, the JFNK system with second order preconditioning exhibits scalability approaching optimal.

the scalability of the approach for a range of different reconstruction algorithms and non-linear convergence criteria for the MHD eigensystem given by [29]: by setting the non-linear convergence criteria to  $\|\mathbf{F}(x_k)\|_2 < \epsilon_a + \epsilon_r \|\mathbf{F}(x_0)\|_2$  while at each outer Newton iteration setting the Krylov solver convergence criteria to  $\left\| \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right\|_k + \|\mathbf{F}(x_k)\|_2 < \zeta_k \|\mathbf{F}(x_0)\|_2$ , we are able to obtain an algorithm that approaches optimal for compressible MHD.

**3.3. Physics-Based Preconditioners for the Adiabatic Two-Fluid Plasma Equations.** Having demonstrated the feasibility of our approach for preconditioning the compressible MHD equations, we now extend this approach to electron-ion fluid plasmas with full-wave electromagnetics. The evolution of the two species are described by:

$$\frac{\partial \rho_s}{\partial t} + \nabla \cdot (\rho_s \mathbf{v}_s) = 0 \quad (3.23)$$

$$\begin{aligned} \frac{\partial(\rho_s \mathbf{v}_s)}{\partial t} + \nabla \cdot (\rho_s \mathbf{v}_s \mathbf{v}_s + \mathbf{I} p_s + \Pi_s) \\ = n_s q_s \mathbf{E} + \mathbf{J}_s \times \mathbf{B} - \sum_{i \neq s} \rho_s \rho_i \lambda_{s,i} (\mathbf{v}_s - \mathbf{v}_i) \end{aligned} \quad (3.24)$$

$$\begin{aligned} \frac{\partial e_s}{\partial t} + \nabla \cdot ((e_s + p_s) \mathbf{v}_s + \mathbf{v}_s \cdot \boldsymbol{\tau}_s + \mathbf{q}_s) \\ = \mathbf{E} \cdot \mathbf{J}_s - \sum_{i \neq s} \rho_s \rho_i \lambda_{s,i} (\mathbf{v}_s - \mathbf{v}_i) \frac{(m_s \mathbf{v}_s - m_i \mathbf{v}_i)}{(m_s - m_i)} \end{aligned} \quad (3.25)$$

where  $s = i, e$  denotes the ‘species’ (e.g. ions, electrons),  $\rho_s$  is the species density,  $\mathbf{v}_s$  is the species velocity and  $e_s$  is the total energy for the species. These equations are coupled through collision operators (e.g.  $\sum_{i \neq s} \rho_s \rho_i \lambda_{s,i} (\mathbf{v}_s - \mathbf{v}_i)$  in eqn. 3.24) and

through the electromagnetic fields, described by Maxwell's equations:

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \quad (3.26)$$

$$\frac{\partial \mathbf{E}}{\partial t} - c^2 \nabla \times \mathbf{B} = \mu_0 \mathbf{J} \quad (3.27)$$

In order to make use of the physics-based preconditioner developed in §3.2, we reformulate the two fluid plasma model in terms of the center-of-mass total density, total momentum density, total charge density and total current density, defined as:

$$\begin{aligned} \rho &= \sum_s \rho_s; \quad \rho \mathbf{v} = \sum_s \rho_s \mathbf{v}_s \\ \rho_c &= \sum_s q_s m_s^{-1} \rho_s; \quad \mathbf{J} = \sum_s q_s m_s^{-1} \rho_s \mathbf{v}_s \end{aligned} \quad (3.28)$$

With these definitions, we are able to write the two fluid plasma equations in MHD-like form:

$$\begin{aligned} U &= \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ E \\ \rho_c \\ \mathbf{J} \end{bmatrix}; \quad F(U) = \begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \mathbf{v}^T + p \mathbb{I} \\ (E + p) \mathbf{v} \\ \mathbf{J} \\ \sum_s \frac{q_s}{m_s} \nabla \cdot (\rho_s \mathbf{v}_s \mathbf{v}_s + \mathbf{I} \rho_s) \end{bmatrix} \\ S(U) &= \begin{bmatrix} 0 \\ \rho_c \mathbf{E} + \mathbf{J} \times \mathbf{B} \\ \mathbf{E} \cdot \mathbf{J} \\ 0 \\ \sum_s \frac{q_s}{m_s} (n_s q_s \mathbf{E} + \mathbf{J}_s \times \mathbf{B} - \sum_{i \neq s} \rho_s \rho_i \lambda_{s,i} (\mathbf{v}_s - \mathbf{v}_i)) \end{bmatrix} \end{aligned} \quad (3.29)$$

where the evolution of the electromagnetic fields are described by Maxwell's equations (3.26,3.27).

The next step in reformulation of the two fluid equations to make the amenable to preconditioning using the MHD eigensystem is to reformulate the source terms for the total momentum and energy equations:

$$S(\rho \mathbf{v}) = \rho_c \mathbf{E} + \mathbf{J} \times \mathbf{B}; \quad S(E) = \mathbf{E} \cdot \mathbf{J} \quad (3.30)$$

These can be rewritten in terms of conservation of electromagnetic momentum and energy, using the standard identities (see, e.g., Jackson, 1975, [34]):

$$\rho_c \mathbf{E} + \mathbf{J} \times \mathbf{B} = -\frac{1}{c^2} \frac{\partial \mathbf{S}_{EM}}{\partial t} + \nabla \cdot \tilde{\mathbf{T}}_{EM} \quad (3.31)$$

$$\mathbf{E} \cdot \mathbf{J} = -\frac{\partial E_{EM}}{\partial t} - \nabla \cdot \mathbf{S}_{EM} \quad (3.32)$$

where:

$$\mathbf{S}_{EM} = \frac{\mathbf{E} \times \mathbf{B}}{\mu_0} \quad (3.33)$$

$$E_{EM} = \frac{1}{2\mu_0} \left( \frac{\mathbf{E} \cdot \mathbf{E}}{c^2} + \mathbf{B} \cdot \mathbf{B} \right) \quad (3.34)$$

$$\tilde{\mathbf{T}}_{EM} = \frac{1}{\mu_0} \left( \frac{\mathbf{E} \mathbf{E}^T}{c^2} + \mathbf{B} \mathbf{B}^T + E_{EM} \mathbb{I} \right) \quad (3.35)$$

where  $E_{EM}$  is the energy density in the electromagnetic field,  $\mathbf{S}_{EM}$  is the Poynting flux and  $\tilde{T}_{EM}$  is the electromagnetic stress tensor. The remaining source term is associated with the evolution of the total current:

$$S(\mathbf{J}) = \sum_s \frac{q_s}{m_s} \left( n_s q_s \mathbf{E} + \mathbf{J}_s \times \mathbf{B} - \sum_{i \neq s} \rho_s \rho_i \lambda_{s,i} (\mathbf{v}_s - \mathbf{v}_i) \right) \quad (3.36)$$

This source term can be made amenable to preconditioning by writing in terms of  $\rho$ ,  $\mathbf{v}$  and  $\mathbf{J}$ :

$$\begin{aligned} S(\mathbf{J}) &= \left( \frac{q_e}{m_i} \right)^2 [(k_\rho \rho + k_{\rho_c} \rho_c) (\mathbf{E} + \eta \mathbf{J}) + (k_\rho \rho \mathbf{v} + k_{\rho_c} \mathbf{J}) \times \mathbf{B}] \\ k_\rho &= \left[ \left( \frac{m_e}{m_i} \right)^2 \frac{q_e}{m_e} - \left( \frac{q_e}{q_i} \right)^2 \frac{q_i}{m_i} \right] \left[ \frac{q_e}{m_e} - \frac{q_i}{m_i} \right]^{-1} \\ k_{\rho_c} &= \left[ \left( \frac{q_e}{q_i} \right)^2 - \left( \frac{m_e}{m_i} \right)^2 \right] \left[ \frac{q_e}{m_e} - \frac{q_i}{m_i} \right]^{-1} \end{aligned} \quad (3.37)$$

The MHD-like two fluid equations written in conservation law form are then:

$$\begin{aligned} U &= \begin{bmatrix} \rho \mathbf{v} + c^{-2} \mathbf{S}_{EM} \\ E + E_{EM} \\ \rho_c \\ \mathbf{J} \end{bmatrix}; \quad F(U) = \begin{bmatrix} \rho \mathbf{v} \mathbf{v}^T + p \mathbb{I} - \tilde{T}_{EM} \\ (E + p) \mathbf{v} + \mathbf{S}_{EM} \\ \mathbf{J} \\ \sum_s \frac{q_s}{m_s} \nabla \cdot (\rho_s \mathbf{v}_s \mathbf{v}_s + \mathbf{I} p_s) \end{bmatrix} \\ S(U) &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \left( \frac{q_e}{m_i} \right)^2 [(k_\rho \rho + k_{\rho_c} \rho_c) (\mathbf{E} + \eta \mathbf{J}) + (k_\rho \rho \mathbf{v} + k_{\rho_c} \mathbf{J}) \times \mathbf{B}] \end{bmatrix} \end{aligned} \quad (3.38)$$

Note that the conserved quantities now include contributions from the electromagnetic field in both the energy and the momentum. We also remark that the contribution of the electromagnetic field to the definition of the total momentum is suppressed by a factor  $c^2$ , while the coupling of the net charge to the electric field has been replaced by contributions  $O(E^2/c^2)$ . As such, the, the equations describing the evolution of total mass, momentum and energy can therefore be preconditioned *neglecting* contributions from the electric field as these can only be important when  $E \sim O(c)$ ; in such a case of a *relativistically* strong electric field, it is inappropriate to use the above set of equations and instead, the equations of relativistic fluid dynamics should be considered (e.g. [35]).

The equations of total charge and current conservation require a separate preconditioning strategy. Here, we utilize the eigensystem for compressible *hydrodynamics* (e.g. [29]) and write the preconditioner for this system as:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}}|_k \delta U_k \approx \left[ 1 - \theta \Delta t \sum_s \frac{q_s}{m_s} \frac{\partial F(U_s)}{\partial U_s} \frac{\partial U_s}{\partial U} + \theta \Delta t \frac{\partial S(U)}{\partial U} \right] \delta U_k \quad (3.39)$$

$$U = \begin{bmatrix} \rho_c \\ \mathbf{J} \end{bmatrix}; \quad U_s = \begin{bmatrix} \rho_s \\ \rho_s \mathbf{v}_s \end{bmatrix}; \quad F(U_s) = \begin{bmatrix} \rho_s \mathbf{v}_s \\ \rho_s \mathbf{v}_s \mathbf{v}_s^T + p_s \mathbb{I} \end{bmatrix} \quad (3.40)$$

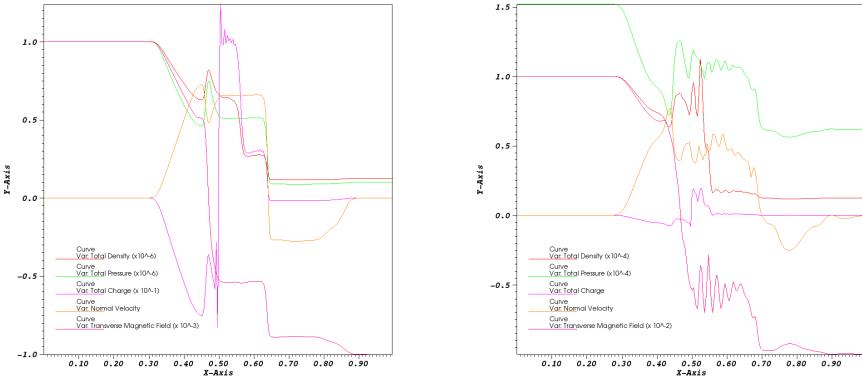


Fig. 3.5: Classic Brio-Wu shock tube in the MHD (left panel) and two-fluid (right panel) regimes, both computed using the fully implicit two-fluid model in MHD-like form. The data of these figures demonstrate the ability of the code to handle shocks across a wide range of parameter regimes.

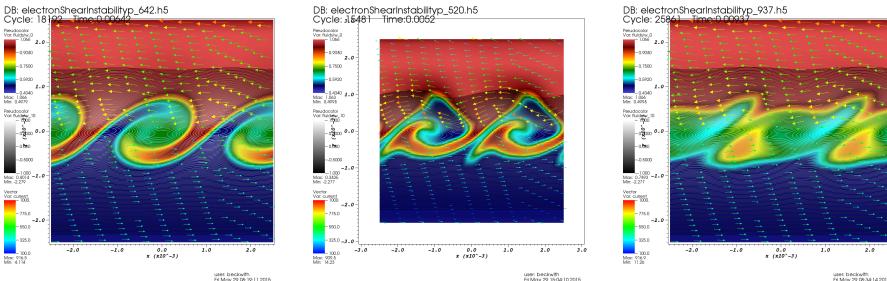


Fig. 3.6: Density, velocity and magnetic field distributions for the evolution of a ion-electron shear instability using the fully implicit two-fluid model in MHD-like form. In this calculation, the ions and electrons are unstable on different timescales, which leads to generation of magnetic islands, demonstrating the ability of the scheme to handle separate ion and electron dynamics.

where  $\partial U_s / \partial U$  can be determined through inspection of eqn. 3.28. In addition, since this system of equations requires knowledge of both the ion and electron pressure separately, we evolve an entropy conservation equation for the electron fluid to enable separation of the ion and electron pressure in the total energy equation. We precondition Maxwell's equations through the eigensystem due to [27], using a Generalized Lagrange Multiplier method described therein, with the addition of a linear hyperbolic system that can be solved directly for the electric field. As for the MHD equations, we precondition the two-fluid system by constructing  $P_k^{-1}$  using Trilinos ML Uncoupled Smoothed Aggregation with 5 levels; on each level, we use block ILU smoothing with zero overlap and symmetric Gauss-Seidel relaxation on each level and choose the block size to be equal to the number of partial differential equations in the system (here, this is 18).

This approach allows us to attack two-fluid plasma problems across a range of regimes within a common solver framework. The data of Fig. 3.5 demonstrates classic one-dimensional shock tubes in both the MHD and two-fluid regimes, where in each case the shock is evolved on the MHD timescale, while the lightwave is many orders of magnitude faster. Next, the data of Fig 3.6 demonstrate the application of this approach to two-fluid shear instabilities where the ions and electrons are unstable on different timescales, which leads to generation of magnetic islands. In both cases, the system was evolved on the MHD timescale, e.g. significantly above the timescales associated with the propagation of electromagnetic waves in the fluid. Taken together, these example problems demonstrate that the MHD-like formulation of the two fluid plasma equations reduces spurious divergence errors in the electric field and can handle separate ion and electron dynamics. Verification and validation work on the two-fluid plasma model is ongoing under AFOSR Grant FA9550-14-C-0004 and will be reported elsewhere [36].

### 3.4. Physics-based Preconditioners for the Isothermal Two-Fluid Plasma

**Equations.** The evolution of a ion-electron isothermal plasma is described by conservation of mass and momentum (e.g. eliminating the energy equation from the compressible system described above):

$$\begin{aligned} m_\alpha \partial_t \mathbf{p}_\alpha + m_\alpha \nabla \cdot \underline{\mathbf{S}}_\alpha - q_\alpha n_\alpha \mathbf{E} - q_\alpha \mathbf{p}_\alpha \times \mathbf{B} &= \mathbf{0}, \\ \partial_t n_\alpha + \nabla \cdot \mathbf{p}_\alpha &= 0, \end{aligned} \quad (3.41)$$

where  $m_\alpha$  is the mass,  $q_\alpha$  is the charge,  $\mathbf{p}_\alpha$  is the momentum density,  $n_\alpha$  is the number density and  $\underline{\mathbf{S}}_\alpha$  is the stress tensor, for species  $\alpha = i, e$ . A simple isothermal closure is supplied:  $(m_\alpha \nabla \cdot \underline{\mathbf{S}}_\alpha) \rightarrow (T_\alpha \nabla n_\alpha) + m_\alpha \nabla \cdot \left( \frac{\mathbf{p}_\alpha \mathbf{p}_\alpha^T}{n_\alpha} \right)$ , with  $T_\alpha$  the temperature. The fluid equations couple directly to Maxwell's field equations,

$$\begin{aligned} \nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon_0}, & (\text{Gauss' Law}) \\ \nabla \times \mathbf{E} + \partial_t \mathbf{B} &= \mathbf{0}, & (\text{Faraday's Law}) \\ \nabla \cdot \mathbf{B} &= 0, & (\text{Solenoidal Constraint}) \\ -\mu_0 \epsilon_0 \partial_t \mathbf{E} + \nabla \times \mathbf{B} &= \mu_0 \mathbf{j}, & (\text{Ampere's Law}) \end{aligned} \quad (3.42)$$

where  $\mathbf{E}$  is the electric field,  $\mathbf{B}$  is the magnetic field,  $\epsilon_0$  is the permittivity,  $\mu_0$  is the permeability,  $\rho$  is the charge density, and  $\mathbf{j}$  is the current density. The charge density and current density are related directly to the number density and momentum density by  $\rho = q_e n_e + q_i n_i$  and  $\mathbf{j} = q_e \mathbf{p}_e + q_i \mathbf{p}_i$ . The time derivatives are implicitly discretized ( $\partial_t \rightarrow \frac{1}{\delta t}$ ), and all known explicit pieces are collected on the right-hand side. Coupling (3.41) with (3.42) produces the first-order system

$$\left[ \begin{array}{c|cc|cc} \frac{m_i}{\delta t} & T_i \nabla & 0 & 0 & -q_i n_i & -q_i \mathbf{p}_i \times \\ \hline \nabla \cdot & \frac{1}{\delta t} & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \frac{m_e}{\delta t} & T_e \nabla & -q_e n_e & -q_e \mathbf{p}_e \times \\ 0 & 0 & \nabla \cdot & \frac{1}{\delta t} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \nabla \times & \frac{1}{\delta t} \\ 0 & -\frac{q_i}{\epsilon_0} & 0 & -\frac{q_e}{\epsilon_0} & \nabla \cdot & 0 \\ -\mu_0 q_i & 0 & -\mu_0 q_e & 0 & -\frac{\mu_0 \epsilon_0}{\delta t} & \nabla \times \\ 0 & 0 & 0 & 0 & 0 & \nabla \cdot \end{array} \right] \begin{bmatrix} \mathbf{p}_i \\ n_i \\ \mathbf{p}_e \\ n_e \\ \mathbf{E} \\ \mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathbf{p}_i} \\ g_{n_i} \\ \mathbf{f}_{\mathbf{p}_e} \\ g_{n_e} \\ \mathbf{f}_{\mathbf{E}} \\ \mathbf{f}_{\mathbf{B}} \end{bmatrix}. \quad (3.43)$$

**3.4.1. Darwin Approximation.** In non-relativistic kinetic simulations the speed of light is spurious and leads to numerical instabilities. Overcoming these instabilities can be accomplished by the injection of artificial dissipation, but only at the expense of energy conservation [37]. These issues can be circumvented by taking the asymptotic limit,  $c \rightarrow \infty$ , or, alternatively,  $\epsilon_0 \rightarrow 0$ . Unfortunately, this rigorously holds true only in the quasineutral approximation. When coupling to a non-relativistic kinetic simulation, this is insufficient; charge separation effects are required. For these reasons, a Darwin model is used. The model is an approximation which decomposes fields into their solenoidal and irrotational components in order to analytically project out light waves while maintaining charge separation in Gauss' law [38], [39]. Without derivation, the Darwin model is (see [40], [39] for a detailed discussion):

$$\begin{aligned} \nabla \times \mathbf{E}_r &= 0, & (\text{Definition}) \\ \nabla \cdot \mathbf{E}_r &= \frac{\rho}{\epsilon_0}, & (\text{Gauss' Law}) \\ -\mu_0 \epsilon_0 \partial_t \mathbf{E}_r + \nabla \times \mathbf{B} &= \mu_0 \mathbf{j}, & (\text{Ampere's law}) \\ \nabla \cdot \mathbf{B} &= 0, & (\text{Solenoidal Constraint}) \\ \partial_t \mathbf{B} + \nabla \times \mathbf{E}_s &= 0, & (\text{Faraday's Law}) \\ \nabla \cdot \mathbf{E}_s &= 0, & (\text{Definition}) \end{aligned} \quad (3.44)$$

where  $\mathbf{E}_r$  is the irrotational electric field and  $\mathbf{E}_s$  is the solenoidal electric field defined such that  $\mathbf{E} = \mathbf{E}_r + \mathbf{E}_s$ . Note that, by definition,  $\mathbf{B}$  is solenoidal. The main difference from Maxwell's equations (3.42) is that  $\partial_t \mathbf{E}_s$  has been removed from Ampere's law.

Coupling (3.44) with (3.41) yields the TFP-Darwin model:

$$\left[ \begin{array}{cc|cc|ccc} \frac{m_i}{\delta t} & T_i \nabla & 0 & 0 & -q_i n_i & -q_i \mathbf{p}_i \times & -q_i n_i \\ \nabla \cdot & \frac{1}{\delta t} & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \frac{m_e}{\delta t} & T_e \nabla & -q_e n_e & -q_e \mathbf{p}_e \times & -q_e n_e \\ 0 & 0 & \nabla \cdot & \frac{1}{\delta t} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \nabla \times & 0 & 0 \\ 0 & -\frac{q_i}{\epsilon_0} & 0 & -\frac{q_e}{\epsilon_0} & \nabla \cdot & 0 & 0 \\ -\mu_0 q_i & 0 & -\mu_0 q_e & 0 & -\frac{\epsilon_0 \mu_0}{\delta t} & \nabla \times & 0 \\ 0 & 0 & 0 & 0 & 0 & \nabla \cdot & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\delta t} & \nabla \times \\ 0 & 0 & 0 & 0 & 0 & 0 & \nabla \cdot \end{array} \right] = \left[ \begin{array}{c} \mathbf{p}_i \\ n_i \\ \hline \mathbf{p}_e \\ n_e \\ \hline \mathbf{E}_r \\ \mathbf{B} \\ \mathbf{E}_s \end{array} \right] = \left[ \begin{array}{c} \mathbf{f}_{\mathbf{p}_i} \\ g_{n_i} \\ \hline \mathbf{f}_{\mathbf{p}_e} \\ g_{n_e} \\ \hline 0 \\ g_{\mathbf{E}_r} \\ g_{\mathbf{B}} \\ 0 \\ \hline \mathbf{f}_{\mathbf{E}_s} \\ 0 \end{array} \right], \quad (3.45)$$

subject to pure conductive boundary conditions,

$$\begin{aligned} \vec{n} \cdot \nabla n_\alpha &= 0, \\ \mathbf{p}_\alpha &= \mathbf{0}, \\ \vec{n} \times (\mathbf{E}_r + \mathbf{E}_s) &= \mathbf{0}, \\ \vec{n} \cdot \mathbf{B} &= 0, \end{aligned} \quad (3.46)$$

where  $\vec{n}$  denotes the outward boundary normal. To make the system (3.45) well posed, the following, stronger boundary conditions are imposed. The additive constraint is decomposed into separate constraints for both the irrotational and solenoidal components of the electric field,

$$\begin{aligned} \vec{n} \times \mathbf{E}_r &= \mathbf{0}, \\ \vec{n} \times \mathbf{E}_s &= \mathbf{0}. \end{aligned} \quad (3.47)$$

**3.4.2. FOSLS-TFP-Darwin.** In this section, each block of the TFP-Darwin system is addressed. The block is scaled to ensure optimal algebraic multigrid (AMG) performance on that block. In the case of the fluid blocks, the system is modified to allow  $\mathcal{H}^1$  ellipticity of that block. Then, the nonlinearities on the full TFP-Darwin model are addressed.

As presented in (3.45), the TFP system does not naturally admit full  $\mathcal{H}^1$ -ellipticity. It can be shown that the system (3.45) is  $\mathcal{V}_0$ -elliptic with  $\mathbf{p}_\alpha \in \mathcal{H}(\text{div})$ ,  $n_\alpha \in \mathcal{H}^1$ , and  $\mathbf{E}_r, \mathbf{B}, \mathbf{E}_s \in (\mathcal{H}^1)^3$ . That is,

$$\mathcal{V}_0 = \mathcal{H}(\text{Div}) \otimes \mathcal{H}^1 \otimes \mathcal{H}(\text{Div}) \otimes \mathcal{H}^1 \otimes (\mathcal{H}^1)^3 \otimes (\mathcal{H}^1)^3 \otimes (\mathcal{H}^1)^3.$$

Posed in this space, momentum densities are not in  $\mathcal{H}^1$ . Special care is required to ensure multigrid algorithms yield optimal results. One approach is to use  $\mathcal{H}(\text{Div})$ -conforming finite elements, for example Raviar-Thomas elements, for the momentum densities and use a special multigrid algorithm that employs distributed relaxation based on the support of the divergence-free basis elements [41, 42]. Instead, (3.45) is directly modified by introducing an additional constraint that yields an  $\mathcal{H}^1$ -elliptic system. The assumption that  $\mathbf{p}_\alpha \in \mathcal{H}^1$  is reasonable for most plasma systems of interest.

To accomplish this, denote the differential matrix system in (3.45) by  $\mathcal{L}(\mathbf{u}) = \mathbf{f}$  and define the block differential matrix systems as

$$\mathcal{L}(\mathbf{u}) = \begin{bmatrix} \mathcal{A}_i & 0 & 0 \\ 0 & \mathcal{A}_e & 0 \\ \mathcal{L}_i & \mathcal{L}_e & \mathcal{D} \end{bmatrix} \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_e \\ \mathbf{u}_D \end{bmatrix} + \begin{bmatrix} \mathcal{U}_i \\ \mathcal{U}_e \\ 0 \end{bmatrix}, \quad (3.48)$$

where the first term is linear and  $\mathcal{U}_i$  and  $\mathcal{U}_e$  contain the quadratic terms that appear in the upper right-hand blocks in (3.45).

Below, each block is examined. First, the Darwin block,  $\mathcal{D}$ , is scaled and shown to be  $\mathcal{H}^1$  elliptic. Then, the fluid blocks,  $\mathcal{A}_i$  and  $\mathcal{A}_e$ , are modified, scaled and shown to  $\mathcal{H}^1$  elliptic. Finally, the Fréchet derivative of the entire system is addressed. In Section 3.4.6 the linearized system is shown to be uniformly  $\mathcal{H}^1$  elliptic.

**3.4.3. Darwin Block.** The Darwin block,  $\mathcal{D}$ , is naturally  $\mathcal{H}^1$ -elliptic, but contains scalings that are dependent upon physical constants. The isolated (uncoupled) Darwin system is

$$\begin{bmatrix} \nabla \times & 0 & 0 \\ \nabla \cdot & 0 & 0 \\ -\frac{\epsilon_0 \mu_0}{\delta t} & \nabla \times & 0 \\ 0 & \nabla \cdot & 0 \\ 0 & \frac{1}{\delta t} & \nabla \times \\ 0 & 0 & \nabla \cdot \end{bmatrix} \begin{bmatrix} \mathbf{E}_r \\ \mathbf{B} \\ \mathbf{E}_s \end{bmatrix} = \begin{bmatrix} 0 \\ g_{\mathbf{E}_r} \\ g_{\mathbf{B}} \\ 0 \\ \mathbf{f}_{\mathbf{E}_s} \\ 0 \end{bmatrix}. \quad (3.49)$$

The formal normal of  $\mathcal{D}$  is,

$$\mathcal{D}^* \mathcal{D} = \begin{bmatrix} \frac{\epsilon_0^2 \mu_0^2}{\delta t} - \Delta & -\frac{\epsilon_0 \mu_0}{\delta t} \nabla \times & 0 \\ -\frac{\epsilon_0 \mu_0}{\delta t} \nabla \times & \frac{1}{\delta t^2} - \Delta & \frac{1}{\delta t} \nabla \times \\ 0 & \frac{1}{\delta t} \nabla \times & -\Delta \end{bmatrix} \quad (3.50)$$

Introducing a left block scaling of  $\text{diag} \left[ I, I, \frac{\delta t}{\epsilon_0 \mu_0} I, \frac{\delta t}{\epsilon_0 \mu_0} I, \frac{\delta t^2}{\epsilon_0 \mu_0} I, \frac{\delta t^2}{\epsilon_0 \mu_0} I \right]$  and a right scaling of  $\text{diag} \left[ I, \frac{\epsilon_0 \mu_0}{\delta t} I, \frac{\epsilon_0 \mu_0}{\delta t^2} I \right]$  fully eliminates all physical constants from  $\mathcal{D}$ . These modifications generate the new system

$$\begin{bmatrix} \nabla \times & 0 & 0 \\ \nabla \cdot & 0 & 0 \\ -1 & \nabla \times & 0 \\ 0 & \nabla \cdot & 0 \\ 0 & 1 & \nabla \times \\ 0 & 0 & \nabla \cdot \end{bmatrix} \begin{bmatrix} \mathbf{E}_r \\ \hat{\mathbf{B}} \\ \hat{\mathbf{E}}_s \end{bmatrix} = \begin{bmatrix} 0 \\ g_{\mathbf{E}_r} \\ \hat{\mathbf{f}}_{\mathbf{B}} \\ 0 \\ \hat{\mathbf{f}}_{\mathbf{E}_s} \\ 0 \end{bmatrix}, \quad (3.51)$$

where the transformed unknowns are defined as:  $\hat{\mathbf{B}} = \frac{\delta t}{\epsilon_0 \mu_0} \mathbf{B}$ , and  $\hat{\mathbf{E}}_s = \frac{\delta t^2}{\epsilon_0 \mu_0} \mathbf{E}_s$ . The right-hand side terms are transformed in a similar way. Identify the transformed Darwin block as  $\hat{\mathcal{D}}$ . With standard boundary conditions, it is straightforward to prove that this system is  $\mathcal{H}^1$  elliptic.

To see this, consider the formal normal of  $\hat{\mathcal{D}}$ ,

$$\hat{\mathcal{D}}^* \hat{\mathcal{D}} = \begin{bmatrix} \mathcal{I} - \Delta & -\nabla \times & 0 \\ -\nabla \times & \mathcal{I} - \Delta & \nabla \times \\ 0 & \nabla \times & -\Delta \end{bmatrix}. \quad (3.52)$$

| Lev. | $\rho(\mathcal{D})$ | $\rho(\hat{\mathcal{D}})$ |
|------|---------------------|---------------------------|
| 1    | 0.56                | 0.048                     |
| 2    | 0.60                | 0.044                     |
| 3    | 0.54                | 0.052                     |
| 4    | 0.50                | 0.049                     |
| 5    | 0.53                | 0.050                     |
| 6    | 0.53                | 0.077                     |

Table 3.1: AMG convergence factors for both the original Darwin system (3.49) with a time step of  $\delta t = \frac{2\pi}{\omega_{p,e}}$  and the scaled Darwin system (3.51) using quadratic finite elements. Level 1 is a  $4 \times 4$  quadrilateral mesh and is refined uniformly 6 times using NI.

This system is differentially diagonally dominant, plus the diagonal blocks dominate the off diagonal blocks.

To demonstrate how algebraic multigrid (AMG) applied to this subsystem will behave, let  $\rho(\mathcal{Q})$  denote the asymptotic convergence factor of AMG applied to a discrete form of the operator  $\mathcal{Q}$ . The value of  $\rho(\mathcal{Q})$  is computed by allowing AMG V-cycles, on each level of refinement to continue until a stable factor is reached. The asymptotic AMG convergence factors for both  $\mathcal{D}$  and  $\hat{\mathcal{D}}$ , discretized using FOSLS, are seen in Table 3.1, where a time step of  $\delta t = \frac{2\pi}{\omega_{p,e}}$  was used in the unscaled block. The scaled Darwin block performs significantly better than the unscaled Darwin block.

**3.4.4. Fluid Blocks.** The fluid blocks,  $\mathcal{A}_\alpha$ , are not naturally  $\mathcal{H}^1$ -elliptic. The root cause of this is due to the fact that there is no curl term associated with the momentum densities,  $\mathbf{p}_\alpha$ . The isolated fluid block is

$$\begin{bmatrix} \frac{m_\alpha}{\delta t} & T_\alpha \nabla \\ \nabla \cdot & \frac{1}{\delta t} \end{bmatrix} \begin{bmatrix} \mathbf{p}_\alpha \\ n_\alpha \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathbf{p}_\alpha} \\ g_{n_\alpha} \end{bmatrix}. \quad (3.53)$$

This system is modified by introducing the Curl of conservation of momentum as an additional equation,

$$\begin{bmatrix} \frac{m_\alpha}{\delta t} & T_\alpha \nabla \\ \frac{m_\alpha}{\delta t} \nabla \times & 0 \\ \nabla \cdot & \frac{1}{\delta t} \end{bmatrix} \begin{bmatrix} \mathbf{p}_\alpha \\ n_\alpha \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathbf{p}_\alpha} \\ \nabla \times \mathbf{f}_{\mathbf{p}_\alpha} \\ g_{n_\alpha} \end{bmatrix}. \quad (3.54)$$

Denote the differential matrix above by  $\mathcal{C}_\alpha$ . The modified system,  $\mathcal{C}_\alpha$ , can easily be shown to be fully  $\mathcal{H}^1$ -elliptic with appropriate boundary conditions. The additional Curl constraint forces the momentum density to be in the smoother space,  $\mathcal{H}^1$ . This is a reasonable requirement to enforce on momentum density. The formal normal of  $\mathcal{C}_\alpha$  is

$$\mathcal{C}^* \mathcal{C} = \begin{bmatrix} \frac{m_\alpha^2}{\delta t^2} (\mathcal{I} - \Delta) + \left( \frac{m_\alpha^2}{\delta t^2} - 1 \right) \nabla \nabla \cdot & \left( \frac{m_\alpha T_\alpha}{\delta t} - \frac{1}{\delta t} \right) \nabla \\ - \left( \frac{m_\alpha T_\alpha}{\delta t} - \frac{1}{\delta t} \right) \nabla \cdot & \frac{1}{\delta t^2} - T_\alpha^2 \Delta \end{bmatrix}. \quad (3.55)$$

Notice that the upper left block may become dominated by the  $-\nabla \nabla \cdot$  term, depending on the size of the physical parameters. Introducing a left block scaling of  $\text{diag} [I, \delta t \sqrt{\frac{T_\alpha}{m_\alpha}} I, \sqrt{m_\alpha T_\alpha} I]$  and a right block scaling of  $\text{diag} [\frac{1}{\sqrt{m_\alpha T_\alpha}} I, \frac{1}{T_\alpha} I]$  yields

| Lev. | $\rho(\mathcal{C}_i)$ | $\rho(\hat{\mathcal{C}}_i)$ | $\rho(\mathcal{C}_e)$ | $\rho(\hat{\mathcal{C}}_e)$ |
|------|-----------------------|-----------------------------|-----------------------|-----------------------------|
| 1    | 0.47                  | 0.054                       | 0.87                  | 0.11                        |
| 2    | 0.73                  | 0.084                       | 0.97                  | 0.13                        |
| 3    | 0.76                  | 0.12                        | 0.99                  | 0.14                        |
| 4    | 0.81                  | 0.13                        | 0.99                  | 0.16                        |
| 5    | 0.87                  | 0.15                        | -                     | 0.16                        |
| 6    | 0.90                  | 0.14                        | -                     | 0.17                        |

Table 3.2: AMG convergence factors for both the original fluid system (3.54) and the scaled fluid system (3.56) using quadratic finite elements. Level 1 is a  $4 \times 4$  quadrilateral mesh refined uniformly 6 times using NI.

the system

$$\begin{bmatrix} \frac{\sqrt{m_\alpha}}{\delta t \sqrt{T_\alpha}} & \nabla \\ \nabla \times & 0 \\ \nabla \cdot & \frac{\sqrt{m_\alpha}}{\delta t \sqrt{T_\alpha}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}}_\alpha \\ \hat{n}_\alpha \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}}_{\mathbf{p}_\alpha} \\ \nabla \times \hat{\mathbf{f}}_{\mathbf{p}_\alpha} \\ g_{n_\alpha} \end{bmatrix}, \quad (3.56)$$

where  $\hat{\mathbf{p}}_\alpha = \sqrt{m_\alpha T_\alpha} \mathbf{p}_\alpha$  and  $\hat{n}_\alpha = T_\alpha n_\alpha$ . The right-hand side terms are transformed in a similar way. Define the differential matrix in (3.56) as  $\hat{\mathcal{C}}_\alpha$ . The formal normal is

$$\hat{\mathcal{C}}_\alpha^* \hat{\mathcal{C}}_\alpha = \begin{bmatrix} \frac{m_i}{T_i \delta t^2} - \Delta & 0 \\ 0 & \frac{m_i}{T_i \delta t^2} - \Delta \end{bmatrix}. \quad (3.57)$$

The off diagonal blocks in the formal normal have been completely eliminated, producing an  $\mathcal{H}^1$ -elliptic block diagonal system. That is, multigrid cycles will be optimal for (3.56) when discretized using FOSLS.

The asymptotic AMG convergence factors for  $\mathcal{C}_i$ ,  $\hat{\mathcal{C}}_i$ ,  $\mathcal{C}_e$ , and  $\hat{\mathcal{C}}_e$  are displayed in Table 3.2. A time step of  $\delta t = \frac{2\pi}{\omega_{p,e}}$  was used. Both the scaled ion and electron fluid blocks show significant improvement over their unscaled counterparts. The unscaled electron fluid block benefits the most from scaling due to the stiffness of the electron mass. The original fluid blocks,  $\mathcal{A}_\alpha$ , are not  $\mathcal{H}^1$  elliptic and standard AMG performs poorly on their discrete systems.

**3.4.5. Full TFP System.** Following the scalings and modifications outlined in Section 3.4.3 and Section 3.4.4 through the entire system, (3.45) yields

$$\tilde{\mathcal{L}}(\hat{\mathbf{u}}) = \begin{bmatrix} \hat{\mathcal{C}}_i & 0 & 0 \\ 0 & \hat{\mathcal{C}}_e & 0 \\ \hat{\mathcal{L}}_i & \hat{\mathcal{L}}_e & \hat{\mathcal{D}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}_i \\ \hat{\mathbf{u}}_e \\ \hat{\mathbf{u}}_D \end{bmatrix} + \begin{bmatrix} \hat{\mathcal{U}}_i \\ \hat{\mathcal{U}}_e \\ 0 \end{bmatrix}, \quad (3.58)$$

where  $\hat{\mathcal{C}}_\alpha$  and  $\hat{\mathcal{D}}$  are defined previously. The scaled off diagonal blocks are

$$\hat{\mathcal{L}}_\alpha = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{q_\alpha}{\epsilon_0 T_\alpha} \\ -\frac{q_\alpha \delta t}{\epsilon_0 \sqrt{m_\alpha T_\alpha}} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.59)$$

and

$$\hat{\mathcal{U}}_\alpha = \begin{bmatrix} -q_\alpha \hat{n}_\alpha \hat{\mathbf{E}}_r - \frac{q_\alpha \epsilon_0 \mu_0}{\delta t} \hat{\mathbf{P}}_\alpha \times \hat{\mathbf{B}} - \frac{q_\alpha \epsilon_0 \mu_0}{\delta t^2} \hat{n}_\alpha \hat{\mathbf{E}}_s \\ -\frac{\delta t q_\alpha \sqrt{T_\alpha}}{\sqrt{m_\alpha}} (\nabla \hat{n}_\alpha) \times \hat{\mathbf{E}}_r - \frac{q_\alpha \epsilon_0 \mu_0 \sqrt{T_\alpha}}{\sqrt{m_\alpha}} \nabla \times (\hat{\mathbf{p}}_\alpha \times \hat{\mathbf{B}}) - \frac{q_\alpha \epsilon_0 \mu_0 \sqrt{T_\alpha}}{\delta t \sqrt{m_\alpha}} \nabla \times (\hat{n}_\alpha \hat{\mathbf{E}}_s) \end{bmatrix}, \quad (3.60)$$

for  $\alpha = i, e$ . The unknowns and right-hand-side functions are also scaled appropriately as outlined in previous sections.

The Fréchet derivative used in the Newton step becomes

$$\tilde{\mathcal{L}}'(\hat{\mathbf{u}})[\delta] = \begin{bmatrix} \hat{\mathcal{C}}_i & 0 & 0 \\ 0 & \hat{\mathcal{C}}_e & 0 \\ \hat{\mathcal{L}}_i & \hat{\mathcal{L}}_e & \hat{\mathcal{D}} \end{bmatrix} \begin{bmatrix} \delta_i \\ \delta_e \\ \delta_D \end{bmatrix} + \begin{bmatrix} \hat{\mathcal{U}}'_{ii} & 0 & \hat{\mathcal{U}}'_{iD} \\ 0 & \hat{\mathcal{U}}'_{ee} & \hat{\mathcal{U}}'_{eD} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_i \\ \delta_e \\ \delta_D \end{bmatrix}, \quad (3.61)$$

where

$$\begin{aligned} \hat{\mathcal{U}}'_{\alpha\alpha} &= \begin{bmatrix} \frac{q_\alpha \epsilon_0 \mu_0}{\delta t} \hat{\mathbf{B}} \times & -q_\alpha \hat{\mathbf{E}}_r - \frac{q_\alpha \epsilon_0 \mu_0}{\delta t^2} \hat{\mathbf{E}}_s \\ \frac{q_\alpha \epsilon_0 \mu_0 \sqrt{T_\alpha}}{\sqrt{m_\alpha}} \nabla \times (\hat{\mathbf{B}} \times & \frac{\delta t q_\alpha \sqrt{T_\alpha}}{\sqrt{m_\alpha}} \hat{\mathbf{E}}_r \times (\nabla - \frac{q_\alpha \epsilon_0 \mu_0 \sqrt{T_\alpha}}{\delta t \sqrt{m_\alpha}} \nabla \times (\hat{\mathbf{E}}_s \\ 0 & 0 \end{bmatrix}, \\ \hat{\mathcal{U}}'_{\alpha D} &= \begin{bmatrix} -q_\alpha \hat{n}_\alpha & -\frac{q_\alpha \epsilon_0 \mu_0}{\delta t} (\hat{\mathbf{p}}_\alpha \times & -\frac{q_\alpha \epsilon_0 \mu_0}{\delta t^2} \hat{n}_\alpha \\ -\frac{\delta t q_\alpha \sqrt{T_\alpha}}{\sqrt{m_\alpha}} (\nabla \hat{n}_\alpha \times & -\frac{q_\alpha \epsilon_0 \mu_0 \sqrt{T_\alpha}}{\sqrt{m_\alpha}} \nabla \times (\hat{\mathbf{p}}_\alpha \times & -\frac{q_\alpha \epsilon_0 \mu_0 \sqrt{T_\alpha}}{\delta t \sqrt{m_\alpha}} \nabla \times (\hat{n}_\alpha \end{bmatrix}, \end{aligned}$$

for  $\alpha = i, e$ , and open parentheses indicate an operation to be taken first.

A final scaling is performed on this system in order to balance the magnitude of the upper and lower off-diagonal blocks without altering the modifications made to the block diagonals<sup>1</sup>. This reduces the size of the off-diagonal terms in the formal normal. Let

$$\alpha = \max \left( 1, \frac{\epsilon_0 \mu_0}{\delta t}, \frac{\epsilon_0 \mu_0}{\delta t^2}, \delta t \sqrt{\frac{T_i}{m_i}}, \epsilon_0 \mu_0 \sqrt{\frac{T_i}{m_i}}, \frac{\epsilon_0 \mu_0}{\delta t} \sqrt{\frac{T_e}{m_e}}, \epsilon_0 \mu_0 \sqrt{\frac{T_e}{m_e}}, \frac{\epsilon_0 \mu_0}{\delta t} \sqrt{\frac{T_e}{m_e}} \right)$$

and

$$\beta = \max \left( \frac{\delta t}{\epsilon_0 \sqrt{m_i, T_i}}, \frac{1}{\epsilon_0 T_i}, \frac{\delta t}{\epsilon_0 \sqrt{m_e, T_e}}, \frac{1}{\epsilon_0 T_e} \right).$$

The final system involves the block scaling

$$\begin{aligned} \hat{\mathcal{L}}'(\hat{\mathbf{u}})[\hat{\delta}] &= \begin{bmatrix} \frac{1}{\sqrt{\alpha}} I & 0 & 0 \\ 0 & \frac{1}{\sqrt{\alpha}} I & 0 \\ 0 & 0 & \frac{1}{\sqrt{\beta}} I \end{bmatrix} \begin{bmatrix} \hat{\mathcal{C}}_i + \hat{\mathcal{U}}'_{ii} & 0 & \hat{\mathcal{U}}'_{iD} \\ 0 & \hat{\mathcal{C}}_e + \hat{\mathcal{U}}'_{ee} & \hat{\mathcal{U}}'_{eD} \\ \hat{\mathcal{L}}_i & \hat{\mathcal{L}}_e & \hat{\mathcal{D}} \end{bmatrix} \begin{bmatrix} \sqrt{\alpha} I & 0 & 0 \\ 0 & \sqrt{\alpha} I & 0 \\ 0 & 0 & \sqrt{\beta} I \end{bmatrix} \begin{bmatrix} \hat{\delta}_i \\ \hat{\delta}_e \\ \hat{\delta}_D \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathcal{C}}_i + \hat{\mathcal{U}}'_{ii} & 0 & \hat{\mathcal{U}}'_{iD} \\ 0 & \hat{\mathcal{C}}_e + \hat{\mathcal{U}}'_{ee} & \hat{\mathcal{U}}'_{eD} \\ \tilde{\mathcal{L}}_i & \tilde{\mathcal{L}}_e & \tilde{\mathcal{D}} \end{bmatrix} \begin{bmatrix} \hat{\delta}_i \\ \hat{\delta}_e \\ \hat{\delta}_D \end{bmatrix}. \end{aligned}$$

<sup>1</sup>It is assumed that the system was previously nondimensionalized and that all constants represent unitless quantities.

**3.4.6. Uniform  $\mathcal{H}^1$  Ellipticity.** An important property of the operator  $\hat{\mathcal{L}}'(\hat{\mathbf{u}})$  is that it be uniformly coercive and continuous in a convenient norm for all  $\hat{\mathbf{u}}$  in some neighborhood of the exact solution,  $\hat{\mathbf{u}}_*$ . This guarantees the existence of each Newton step once the approximation is sufficiently accurate. Below it is shown that, under mild hypotheses,  $\hat{\mathcal{L}}'(\mathbf{u})$  is uniformly  $\mathcal{H}^1$ -elliptic in a neighborhood of the exact solution. Toward that end, define the open ball,

$$B_r(\hat{\mathbf{u}}_*) = \{\hat{\mathbf{u}} \in \mathcal{V} : \|\hat{\mathbf{u}} - \hat{\mathbf{u}}_*\|_{\mathcal{H}^1} < r\}. \quad (3.62)$$

It was shown in Section 3.4.4 that  $\hat{\mathcal{C}}_i$  and  $\hat{\mathcal{C}}_e$  are coercive and continuous in  $\mathcal{H}^1$ . Here, a further assumption is made on the  $\mathcal{H}^1$  ellipticity of the block diagonals,  $\hat{\mathcal{C}}_i + \hat{\mathcal{U}}'_{ii}$  and  $\hat{\mathcal{C}}_e + \hat{\mathcal{U}}'_{ee}$ .

**THEOREM 3.1.** *Let  $\Omega$  be a convex domain with piecewise  $\mathcal{C}^{1,1}$  boundary. Assume that there exist  $r > 0$  and positive constants,  $c_\alpha(r)$  and  $C_\alpha(r)$ , such that, for every  $\hat{\mathbf{u}} \in B_r(\hat{\mathbf{u}}_*)$  and for every  $\hat{\delta}_\alpha \in (\mathcal{H}^1)^4$ ,*

$$c_\alpha(r)\|\hat{\delta}_\alpha\|_{\mathcal{H}^1} \leq \|(\hat{\mathcal{C}}_\alpha + \hat{\mathcal{U}}'_{\alpha\alpha}(\hat{\mathbf{u}}))\hat{\delta}_\alpha\| \leq C_\alpha(r)\|\hat{\delta}_\alpha\|_{\mathcal{H}^1}, \quad (3.63)$$

for  $\alpha = i, e$ . Further, assume that  $\hat{\mathcal{L}}'(\hat{\mathbf{u}})$  is injective for every  $\hat{\mathbf{u}} \in B_r(\hat{\mathbf{u}}_*)$ . Then, there exist positive constants,  $c(r)$  and  $C(r)$ , such that, for every  $\hat{\mathbf{u}} \in B_r(\hat{\mathbf{u}}_*)$  and for every  $\hat{\delta} \in (\mathcal{H}^1)^{17}$ ,

$$c(r)\|\hat{\delta}\|_{\mathcal{H}^1} \leq \|\hat{\mathcal{L}}'(\hat{\mathbf{u}})[\hat{\delta}]\| \leq C(r)\|\hat{\delta}\|_{\mathcal{H}^1}. \quad (3.64)$$

*Proof.* The upper bound in (3.64) is easily obtained and, for the sake of brevity, the proof of the lower bound is only outlined. The ellipticity of  $\hat{\mathcal{D}}$  was established in 3.4.3, with some constants, say  $c_D, C_D > 0$ . Let  $\hat{\mathbf{u}} \in B_r(\hat{\mathbf{u}}_*)$ . By assumption (3.63), the operator with only the block diagonals is  $\mathcal{H}^1$  elliptic with constants  $c_B = \min[c_\alpha(r), c_D]$ ,  $C_B = \max[C_\alpha, C_D]$ .

It is clear that the upper blocks satisfy

$$\|\tilde{\mathcal{U}}_{\alpha D}\hat{\delta}_D\| \leq C_{\alpha D}\|\hat{\delta}_D\|_{\mathcal{H}^1} \leq C_{\alpha D}/c_D\|\hat{\mathcal{D}}\hat{\delta}_D\|, \quad (3.65)$$

for some constants  $C_{\alpha D}(\hat{\mathbf{u}}) > 0$ , for  $\alpha = i, e$ . Now, consider the system with the strictly lower-triangular blocks removed. It is straight forward to establish

$$\left\| \begin{bmatrix} \hat{\mathcal{C}}_i + \hat{\mathcal{U}}'_{ii} & 0 & 0 \\ 0 & \hat{\mathcal{C}}_e + \hat{\mathcal{U}}'_{ee} & 0 \\ 0 & 0 & \hat{\mathcal{D}} \end{bmatrix} \begin{bmatrix} \hat{\delta}_i \\ \hat{\delta}_e \\ \hat{\delta}_D \end{bmatrix} \right\| \leq \eta \left\| \begin{bmatrix} \hat{\mathcal{C}}_i + \hat{\mathcal{U}}'_{ii} & 0 & \tilde{\mathcal{U}}'_{iD} \\ 0 & \hat{\mathcal{C}}_e + \hat{\mathcal{U}}'_{ee} & \tilde{\mathcal{U}}'_{eD} \\ 0 & 0 & \hat{\mathcal{D}} \end{bmatrix} \begin{bmatrix} \hat{\delta}_i \\ \hat{\delta}_e \\ \hat{\delta}_D \end{bmatrix} \right\|,$$

with  $\eta = (1 + \max[C_{\alpha D}]/c_D)$ . This establishes the lower bound for the upper block triangular part of the system. The proof is completed by noting that  $\tilde{\mathcal{L}}_\alpha$  contain only zeroth order terms and by applying a standard compactness argument.  $\square$

**REMARK 3.1.** *It should be clearly asserted that the assumption on the  $\mathcal{H}^1$ -ellipticity of  $(\hat{\mathcal{C}}_\alpha + \hat{\mathcal{U}}'_{\alpha\alpha}(\hat{\mathbf{u}}))$  has not yet been rigorously validated. The numerical tests in Section 3.4.7 provide a strong indication that the above assumption is plausible. A more thorough examination of ellipticity is currently being explored by the authors.*

Full  $\mathcal{H}^1$ -ellipticity implies that standard,  $\mathcal{H}^1$ -conforming, finite element spaces can be used and standard convergence bounds apply. In the linear case, enhanced  $L^2$  convergence is also ensured [43]. That is, the  $L^2$ -norm of the error will converge one order faster than the functional norm of the error. This is observed in the numerical tests below.

**3.4.7. Numerical Tests.** In this section the performance results of the FOSLS-TFP model, solved with NI, are presented. No time stepping algorithm is implemented. Instead, focus is given to the system generated for a fixed time step by manufacturing solutions and analyzing the AMG solver performance and FOSLS convergence properties. Two different problems are analyzed. In the first, a simple smooth solution is constructed from sines and cosines. In the second, a solution is manufactured with a sharp gradient in the  $z$ -component of momentum density and a large circulation in the  $x$  and  $y$  components of the magnetic field. Large features in the fluid variables produce substantial nonlinearities in off diagonal components, as appear in (3.60). It is noted that neither of these manufactured solutions are physical, but instead, are used to demonstrate how the nonlinear solver, nested-iteration, and AMG perform.

The problems are solved on the computational domain  $\Omega = [0, 1] \times [0, 1]$ . The system is nondimensionalized against ion mass,  $m_i$ , fundamental charge,  $e$ , Debye length multiples,  $\ell\lambda_D$ , ion number density,  $n_{0,i}$ , and the electron thermal velocity,  $v_{th,e}$  (Note:  $\ell$  specifies the number of Debye lengths the computational domain represents). For each simulation, the ratios  $\frac{m_e}{m_i}$ ,  $\frac{v_{th,i}}{v_{th,e}}$ ,  $\frac{v_{th,e}}{c}$ , and  $\ell$  must be specified. Unless otherwise denoted, take  $\frac{m_e}{m_i} = 5.44 \times 10^{-4}$ ,  $\frac{v_{th,i}}{v_{th,e}} = 10^{-2}$ ,  $\frac{v_{th,e}}{c} = 10^{-2}$ , and  $\ell = 100$ . In all tests a time step of  $\delta t = 10 \frac{2\pi}{\omega_{p,e}}$  is used.

The coarsest problem, Level 0, is a  $4 \times 4$  mesh discretized with bilinear finite elements. A random initial guess is made and Newton-FOSLS is performed until a desired tolerance is reached. As will be demonstrated, the work performed on Level 0 is negligible. Using  $p$ -refinement, the initial guess for Level 1 is constructed from the approximate solution on Level 0, and standard NI-Newton-FOSLS is continued until the finest level is reached.

**3.4.8. Baseline Test.** In this test a simple smooth solution is constructed from sines and cosines. The number densities are taken to be small perturbations away from 1 and most of the momentum and field quantities are taken to be small perturbations away from 0. The exception is the quantities  $p_{i,z}$ ,  $p_{e,z}$ ,  $B_x$ , and  $B_y$ , which are  $O(1)$ . Prescribe the exact solution to be:

$$\begin{aligned}
p_{i,x,y} &= \frac{1}{10} \sin(\pi x) \sin(\pi y), \\
p_{i,z} &= \sin(\pi x) \sin(\pi y), \\
n_i &= 1 + \frac{1}{10} \cos(2\pi x) \cos(2\pi y), \\
p_{e,x,y} &= -\frac{1}{10} \sin(\pi x) \sin(\pi y), \\
p_{e,z} &= -\sin(\pi x) \sin(\pi y), \\
n_e &= 1 + \frac{1}{10} \cos(2\pi x) \cos(2\pi y), \\
E_{r,x} &= \frac{1}{10} \cos(2\pi x) \sin(\pi y), \\
E_{r,y} &= \frac{1}{10} \sin(\pi x) \cos(2\pi y), \\
E_{r,z} &= \frac{1}{10} \sin(\pi x) \sin(\pi y), \\
B_x &= \sin(\pi x) \cos(2\pi y), \\
B_y &= \cos(2\pi x) \sin(\pi y), \\
B_z &= \frac{1}{10} \cos(2\pi x) \cos(2\pi y), \\
E_{s,x} &= \frac{1}{10} \cos(2\pi x) \sin(\pi y), \\
E_{s,y} &= \frac{1}{10} \sin(\pi x) \cos(2\pi y), \\
E_{s,z} &= \frac{1}{10} \sin(\pi x) \sin(\pi y).
\end{aligned} \tag{3.66}$$

The asymptotic AMG convergence factors,  $\rho$ , for each level are seen in Table 3.3. The factors are computed by setting the AMG solver tolerance to  $10^{-6}$ , effectively allowing

| Lev. | $\rho_1(\mathcal{L})$ | $\rho_2(\mathcal{L})$ |
|------|-----------------------|-----------------------|
| 1    | 0.51                  | 0.57                  |
| 2    | 0.57                  | 0.62                  |
| 3    | 0.49                  | 0.48                  |
| 4    | 0.49                  | 0.52                  |
| 5    | 0.53                  | 0.59                  |
| 6    | 0.59                  | —                     |

Table 3.3: Problem (3.66): Asymptotic AMG convergence factors for the final scaled nonlinear TFP-Darwin system. A relative AMG Solver tolerance of  $10^{-6}$  was used. Level 1 is a  $4 \times 4$  quadrilateral meshed discretized with quadratic elements.  $\rho_i$  denotes the factor produced for Newton iteration  $i$ .

AMG V-cycles to continue until a stable convergence factor is reached. This is done independently from the NI-Newton-FOSLS computations. Several Newton iterations are performed on each level, each with its own asymptotic factor. Denote  $\rho_i$  as the asymptotic convergence for each Newton iteration  $i$ . The factors hold between 0.50 - 0.60. A substantial quantity of Newton iterations are performed on Level 0, and thus, not included in the table. Averaging over all V-cycles for every Newton iteration on Level 0 gives an average asymptotic convergence factor of 0.35.

The normalized nonlinear FOSLS functional values,  $\|\mathcal{L}(u) - f\|_0$ , are seen in Fig. 3.7, where  $u_0$  represents the value before Newton-FOSLS has been performed and  $u_f$  represents the value after convergence is achieved. The values are normalized such that the initial FOSLS functional on Level 1 has a value of 1. Initially, the functional is decreasing at a rate of nearly  $O(h^3)$ , but as refinement continues it begins to approach  $O(h^2)$ . The  $L^2$ -error of the solution after each Newton-FOSLS process is plotted in Fig. 3.8. The values are normalized such that the initial  $L^2$ -error on Level 1 has a value of 1. The error is decreasing at a rate of  $O(h^3)$ . The rate of  $L^2$ -convergence is one order higher than that of the FOSLS functional, as expected, using the well-known Aubin-Nitsche duality argument [43].

The benefit of NI is well demonstrated in Table. 3.4. Define a Work Unit (WU) to be the cost of one matrix-vector multiplication on the finest level. It takes a total of 20.64 WU in order to bring the solution through the coarse grids and solve the nonlinear problem on Level 6. The 8 Newton iterations performed on Level 0 amounts to only 0.05% of the total work. Notice that the number of Newton iterations required on each successive level decreases such that by the time the finest grid is reached only 1 Newton iteration, using 3 AMG V-cycles, is required. In this test, the components of the solution are smooth and, in turn, keep the off diagonal couplings small. Another test problem is used to demonstrate how the Newton-FOSLS-NI approach performs on problems with sharper features (ie. steep gradients).

**3.4.9. Sharp Current Density Test.** In this test all components of the solution remain the same as in problem (3.66) except for  $p_{i,z}$ ,  $p_{e,z}$ ,  $B_x$ , and  $B_y$ . For these

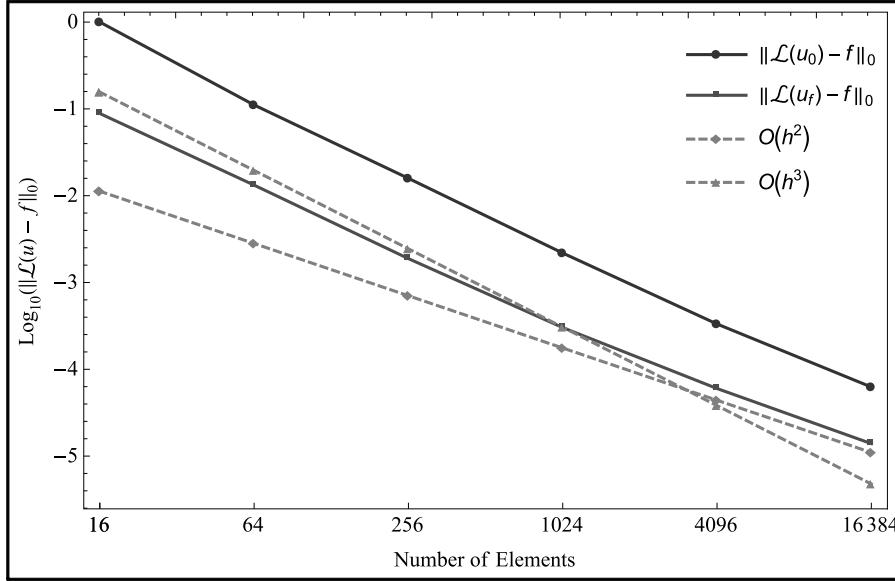


Fig. 3.7: Problem (3.66): The initial and final nonlinear FOSLS functionals through 6 levels of NI with uniform refinement. The tolerance of the AMG Solver is  $10^{-1}$  and the tolerance of the Newton iteration is  $10^{-2}$ . The value of  $h$  is defined as  $\frac{1}{\sqrt{N_e}}$ , where  $N_e$  is the number of elements. The values are normalized such that the initial, nonlinear FOSLS functional on Level 1 has a value of 1.

| Lev.  | Newton Iters. | Tot. V-Cycles | NNZ               | WU    |
|-------|---------------|---------------|-------------------|-------|
| 0     | 8             | 39            | $4.3 \times 10^4$ | 0.01  |
| 1     | 3             | 11            | $2.8 \times 10^5$ | 0.02  |
| 2     | 3             | 13            | $1.1 \times 10^6$ | 0.13  |
| 3     | 3             | 12            | $4.3 \times 10^6$ | 0.61  |
| 4     | 3             | 13            | $1.7 \times 10^7$ | 2.72  |
| 5     | 2             | 9             | $6.8 \times 10^7$ | 7.83  |
| 6     | 1             | 3             | $2.7 \times 10^8$ | 9.31  |
| total | —             | —             | —                 | 20.64 |

Table 3.4: Problem (3.66): The number of Newton Iterations (Newton Iters.), Total number of AMG V-cycles (Tot. V-Cycles), number of nonzeros in the linear operator (NNZ), and Work Units (WU), on each level of the NI process. A WU is defined as the cost of one matrix-vector multiplication on the finest level. The tolerance of the a AMG Solver is  $10^{-1}$  and the tolerance of the Newton iteration is  $10^{-2}$ .

unknowns define:

$$\begin{aligned}
 p_{i,z} &= \sin(\pi x) \sin(\pi y) \exp\left(-\frac{(x-0.5)^2}{2\sigma^2}\right) \exp\left(-\frac{(y-0.5)^2}{2\sigma^2}\right), \\
 p_{e,z} &= -\sin(\pi x) \sin(\pi y) \exp\left(-\frac{(x-0.5)^2}{2\sigma^2}\right) \exp\left(-\frac{(y-0.5)^2}{2\sigma^2}\right), \\
 B_x &= -(y - 0.5) \sin(\pi x) \exp\left(-\frac{(x-0.5)^2}{2\sigma^2}\right) \exp\left(-\frac{(y-0.5)^2}{2\sigma^2}\right), \\
 B_y &= (x - 0.5) \sin(\pi y) \exp\left(-\frac{(x-0.5)^2}{2\sigma^2}\right) \exp\left(-\frac{(y-0.5)^2}{2\sigma^2}\right),
 \end{aligned} \tag{3.67}$$

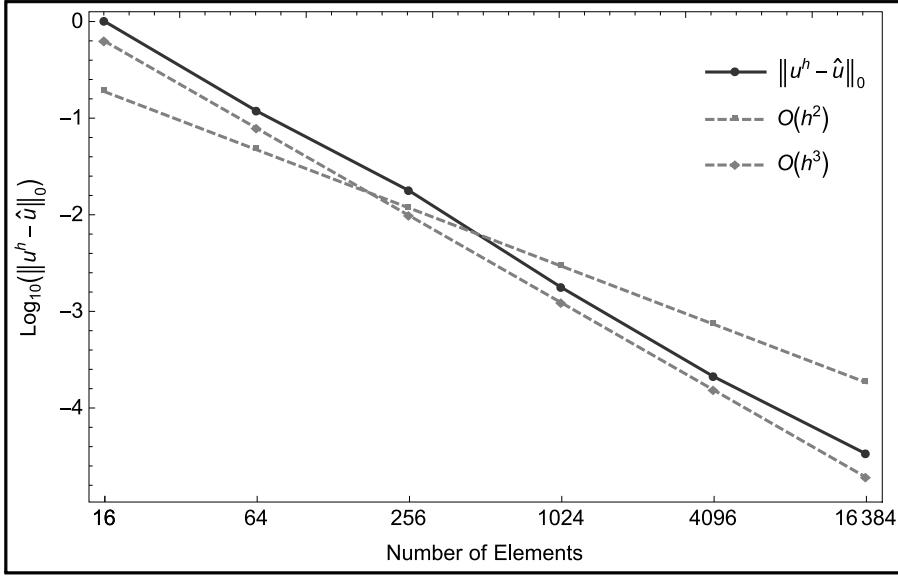


Fig. 3.8: Problem (3.66): The final  $L^2$ -error on each level, through 6 levels of NI with uniform refinement. The tolerance of the AMG Solver is  $10^{-1}$  and the tolerance of the Newton iteration is  $10^{-2}$ . The solutions are normalized such that the error on the Level 1 is 1. Here  $\hat{u}$  represents the interpolant of the exact solution. The value of  $h$  is defined as  $\frac{1}{\sqrt{N_e}}$ , where  $N_e$  is the number of elements.

| Lev. | $\rho_1(\mathcal{L})$ | $\rho_2(\mathcal{L})$ |
|------|-----------------------|-----------------------|
| 1    | 0.54                  | 0.55                  |
| 2    | 0.59                  | .60                   |
| 3    | 0.45                  | —                     |
| 4    | 0.47                  | —                     |
| 5    | 0.55                  | —                     |
| 6    | 0.54                  | —                     |

Table 3.5: Problem (3.67): Asymptotic AMG convergence factors for the final scaled nonlinear TFP-Darwin system. A relative AMG Solver tolerance of  $10^{-6}$  was used. Level 1 is a  $4 \times 4$  quadrilateral meshed discretized with quadratic elements.  $\rho_i$  denotes the factor produced for Newton iteration  $i$ .

with a value of  $\sigma = 0.02$ . The solution contains a current density ( $j_z = q_i p_{i,z} + q_e p_{e,z}$ ) in the  $z$ -direction with a steep gradient and a corresponding strong circulation in the  $(x, y)$ -components of magnetic field. A sketch of these components are visualized in Fig. 3.9.

The asymptotic convergence factors are found in Table 3.5. The factors remain mostly unchanged from problem (3.67), living in the range of 0.5 - 0.6. Level 0 is not included in Table 3.5 because a large number of Newton iterations were performed. The average AMG convergence factor over all V-cycles on Level 0 is 0.45. A slightly tighter relative tolerance of  $10^{-2}$  was required for the AMG solver in order to keep

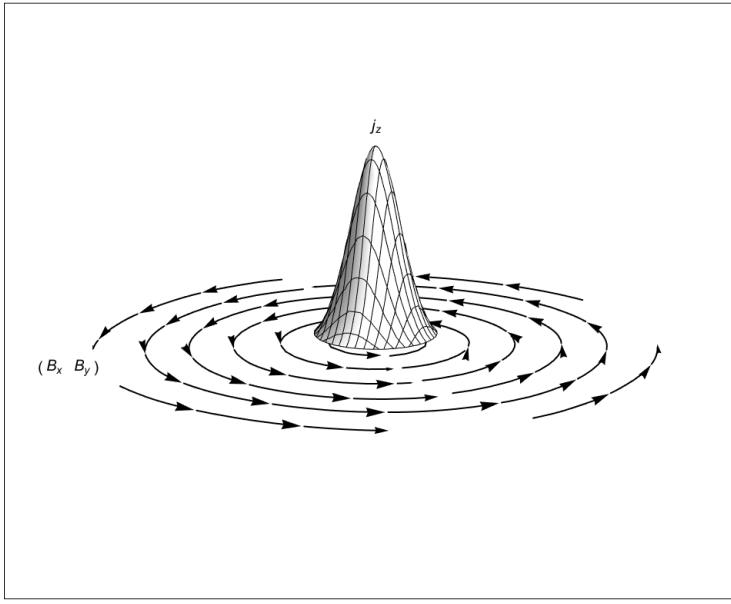


Fig. 3.9: Problem (3.67): A sketch of the  $z$ -component of current density and  $(x, y)$ -components of magnetic field.

the number of Newton iterations low on the finest mesh. The normalized nonlinear FOSLS functional values,  $\|\mathcal{L}(u) - f\|_0$ , are seen in Fig. 3.10, with  $u_0$  and  $u_f$  defined previously. For the coarser grids, the FOSLS functional convergence is poor. This is likely due to the fact that the features present in the current density and magnetic fields are smaller than the resolution of the grid. After 3 levels of  $h$ -refinement (1024 elements) the functional approaches the desired  $O(h^2)$  behavior. The  $L^2$ -error is seen in Fig. 3.11. In a similar way to the nonlinear FOSLS functional, the  $L^2$ -error does not begin to achieve the desired  $O(h^3)$  convergence until 4 levels of  $h$ -refinement are performed. Further refinement would reveal a more conclusive  $O(h^3)$  convergence trend. A summary of the required WUs can be found in Table 3.6. Again, more nonlinear iterations are performed on the coarse grids where V-cycles and Newton linearization are cheap. By the time that the finest level is reached, only 2 Newton iterations and a total of 9 AMG V-cycles were required.

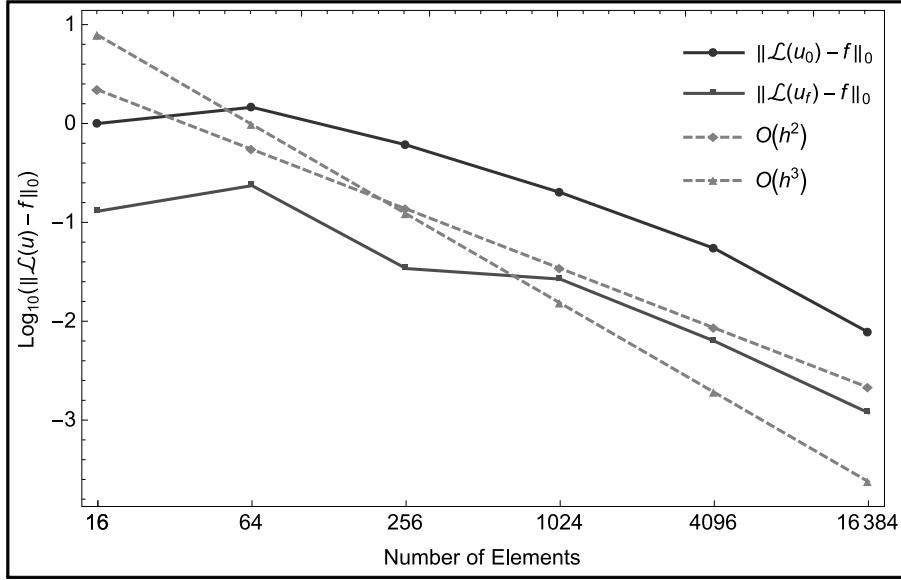


Fig. 3.10: Problem (3.67): The initial and final nonlinear FOSLS functionals through 6 levels of NI with uniform refinement. The tolerance of the a AMG Solver is  $10^{-2}$  and the tolerance of the Newton iteration is  $10^{-2}$ . The value of  $h$  is defined as  $\frac{1}{\sqrt{N_e}}$ , where  $N_e$  is the number of elements. The values are normalized such that the initial nonlinear FOSLS functional on Level 1 has a value of 1.

| Lev.  | Newton Iters. | Tot. V-Cycles | NNZ               | WU    |
|-------|---------------|---------------|-------------------|-------|
| 0     | 7             | 34            | $4.3 \times 10^4$ | 0.01  |
| 1     | 3             | 11            | $2.8 \times 10^5$ | 0.02  |
| 2     | 4             | 17            | $1.1 \times 10^6$ | 0.17  |
| 3     | 1             | 3             | $4.3 \times 10^6$ | 0.13  |
| 4     | 2             | 7             | $1.7 \times 10^7$ | 1.38  |
| 5     | 1             | 3             | $6.8 \times 10^7$ | 2.21  |
| 6     | 2             | 9             | $2.7 \times 10^8$ | 31.33 |
| total | —             | —             | —                 | 35.27 |

Table 3.6: Problem (3.67): The number of Newton Iterations (Newton Iters.), Total number of AMG V-cycles (Tot. V-Cycles), number of nonzeros in the linear operator (NNZ), and Work Units (WU), on each level of the NI process. A WU is defined as the cost of a matrix-vector multiplication on the finest level. The tolerance of the a AMG Solver is  $10^{-2}$  and the tolerance of the Newton iteration is  $10^{-2}$ .

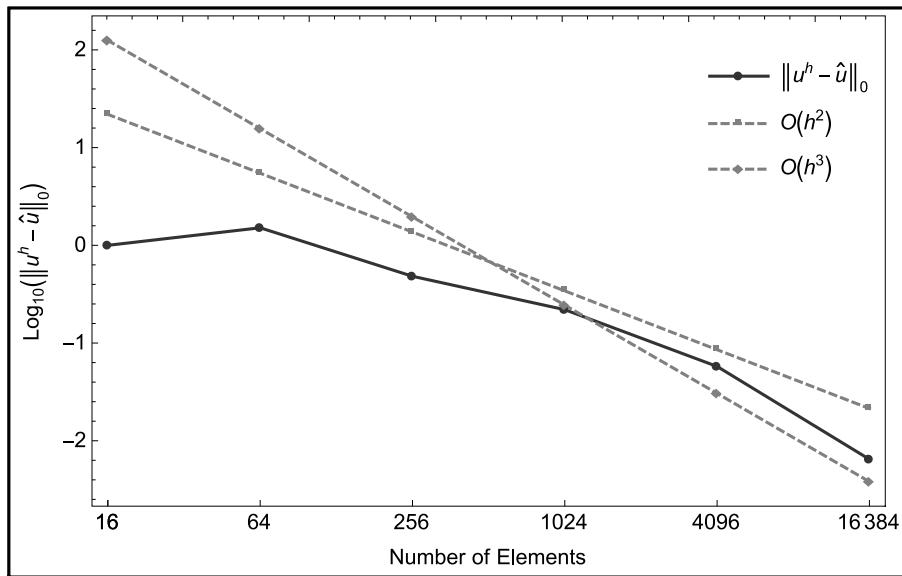


Fig. 3.11: Problem (3.67): The final  $L^2$ -error on each level through 6 levels of NI with uniform refinement. The tolerance of the a AMG Solver is  $10^{-2}$  and the tolerance of the Newton iteration is  $10^{-2}$ . The solutions are normalized such that the error on the Level 0 is 1. Here  $\hat{u}$  represents the interpolant of the exact solution. The value of  $h$  is defined as  $\frac{1}{\sqrt{N_e}}$ , where  $N_e$  is the number of elements.

**4. Space-Time Parallelization.** The need for parallelism in time is being driven by changes in computer architectures, where future speed-ups will be available through greater concurrency, not faster clock speeds. This leads to a bottleneck for sequential time marching schemes because they lack parallelism in the time dimension. Multigrid Reduction in Time (MGRIT [44, 45]) is an iterative procedure that allows for temporal parallelism by utilizing multigrid reduction techniques and a multilevel hierarchy of coarse time grids. The goal of this work is the efficient solution of nonlinear problems with MGRIT, where efficiency is defined as achieving similar performance when compared to an equivalent linear problem. When solving a linear problem using implicit methods and optimal spatial solvers (e.g. classical multigrid), the spatial multigrid convergence rate is fixed across temporal levels, despite a large variation in time step sizes. This is not the case for nonlinear problems, where the work required increases dramatically on coarser time grids. By using a variety of strategies, most importantly, spatial coarsening and an alternate initial guess for the nonlinear solver, it is possible to reduce the work per time step evaluation over all temporal levels to a range similar to those of a corresponding linear problem. This allows for overall speedups comparable with those achieved, in previous work, for linear systems.

Previously, ever increasing clock speeds allowed for the speedup of sequential time integration simulations of a fixed size, and for stable wall clock times for simulations that were refined in space (and usually time). However, clock speeds are now almost stagnant, leading to the sequential time integration bottleneck. By allowing for parallelism in time, much greater computational resources can be brought to bear, and overall speedups can be achieved. Because of this, interest in parallel-in-time methods has grown over the last decade.

Work on parallel-in-time methods actually goes back at least 50 years (c.f. [46] and the review in [47]), but most of the historical efforts were limited to single- or two-level approaches. This project’s focus was on MGRIT, a true multilevel algorithm with optimal scaling in terms of both parallel communication and number of operations. Note that Parareal [48], perhaps the most well known parallel-in-time method, is equivalent [49] to a two-level multigrid scheme.

Sequential time marching schemes are optimal in that they move from the initial time to the final time with minimal computational cost. By applying time stepping on various levels as MGRIT does, some computational cost is traded for efficient temporal concurrency. This efficiency was demonstrated in a strong scaling study of MGRIT for linear diffusion on a  $(257)^2 \times 16385$  space-time grid. The space-time parallel runs used an  $8 \times 8$  processor grid in space, with all additional processors being added in time. MGRIT showed substantially better performance than sequential time stepping when there were more than just a thousand total cores. These strong scaling also showed significantly increased benefits of MGRIT of up to a factor of ten speedup over sequential time stepping in important practical parameter regimes. Moreover, weak scaling results proved MGRIT to be scalable for nonlinear problems, with iteration counts bounded independently of problem size.

**5. Fault Resilience.** As the complexity of high performance computing (HPC) systems continues to progress, constraints on system design force the handling of errors to higher levels in the software stack. Checkpoint restart is the predominate way to recover from fail-stop errors (errors that cause unexpected application termination such as power outage, node reboot, and software crash). Current approaches exploit the complex memory hierarchies of HPC systems and are application tailored for

optimal performance. HPC faces a potentially more harmful class of errors through so-called bit-flips that are caused by a number of external affects such as energetic particle strikes. These are regarded as particularly harmful as they are unexpected by the application and may have silent, unknown impact on the simulation results. This silent data corruption (SDC) can be masked, requiring extra time for a correct simulation, yielding invalid results, or causing application failure. Due to the costly side effects, applications are employing lightweight SDC detectors to verify the correctness of a simulation. For example, Figure 5.1 shows different bits corrupted in the residual calculation of a multigrid cycle, leading to a range of results, from very little impact, to a significant increase in the time-to-solution.

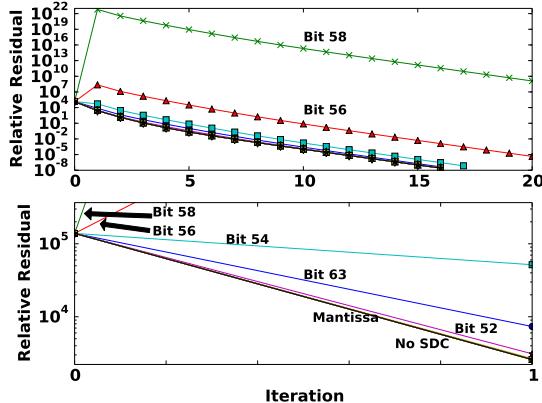


Fig. 5.1: Residual history for different, corrupted bits.

Emerging and next generation architectures are expected to introduce unprecedented levels of faults, requiring new approaches to fault resilience and algorithm development. One difficult aspect in preparing algorithms and application codes for environments with increased levels of SDC is testing. To facilitate better analysis of high-performance scientific libraries, a fault injection and analysis tool called **FlipIt** was created [50]. **FlipIt** uses LLVM to expose intermediate IR code. This IR code is used to create faults in the program logic, thus allowing for general fault models. An example is shown in Figure 5.2 where arithmetic is corrupted through the transformed code. Indeed, in contrast to previous research that focused on faults arising in heavily protected system memory, this package simulates corruption from processor flow, including pointer, arithmetic, and control logic.

The **FlipIt** fault injection framework led to the analysis of several computing kernels, including the sparse matrix-vector product [51]. In addition, a full fault analysis and recovery algorithm was introduced in a multigrid setting [52], where it was shown that the multigrid algorithm may recover from SDC by incorporating several detection strategies. For example, an energy estimate is used within the algorithm to identify errors that may not lead to a segmentation fault, potentially resulting in incorrect or lengthy computations. In addition, simple loop checks were explored as a mechanism for pointer logic that avoids expensive triplication in the code. Finally, the approach introduced a strategy to recover from segmentation faults, thus avoiding an expensive restart if computation could proceed normally. The cost of incorporating

```

define i32 @add(i32 %a, i32 %b) #0 {
entry:
    %add = add nsw i32 %a, %b
    %data = sext i32 %add i64
    %tmp = call i32 @crptInt(i32 0, i32 0,
                           double 0.01, i32 2, i64 %data)
    %crptAdd = trunc i64 %tmp to i32
    ret i32 %crptAdd
}

```

(a) Original LLVM IR.

(b) Transformed LLVM IR.

Fig. 5.2: Intermediate code for fault injection.

these strategies is shown in Figure 5.3 for the case of multigrid. Even in parallel the overhead is modest, while adding a significant amount of protection to the algorithm, leading to nearly 99% successful detection. A clear benefit of the approach is that it is applicable to a wide range of computations.

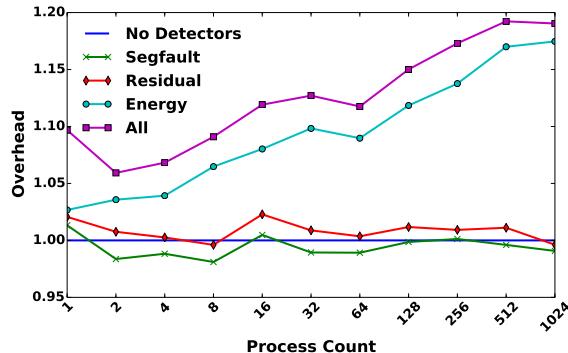


Fig. 5.3: Cost of fault detection versus processor count.

Understanding how (silent) errors propagate through scientific codes is critical, particularly in a high-performance setting. The fault analysis revealed fault patterns across codes that suggest common mitigation strategies [53]. For example, over a large sample of simulations, faults emerge in pointer logic around 40% of the time, arithmetic around 40%, and control logic around 20%. The impact of data corruption on other parts of the application is also important. More recently, the injection approach is being used to study a broader range of methods and applications such as multigrid, FFT, particle methods, plasma computations, and turbulent fluid flow [54]. The approach is leading to visualization tools to help aid developers in identifying sensitive portions of their code.

The fault injection framework and application analysis has also led to a new process for checkpointing use application-level decisions. Checkpointing is critical for recovery from hard faults such as node failure. However, lightweight checkpointing is emerging as a critical tool for recovery when silent errors are detected in the simulation. The fault analysis is leading to the construction of a lossy checkpoint strategy wherein simulation states are checkpointed in-memory. This introduces a small error in comparison to the large decrease in memory (and computational) footprint.

## REFERENCES

- [1] Brandt A, McCormick S, Ruge J. Algebraic multigrid (AMG) for sparse matrix equations. In *Sparsity and its Applications, D.J. Evans (ed.)*, 1984; 257–284.
- [2] Ruge J, Stüben K. *Algebraic multigrid (AMG)*. In *Multigrid Methods, vol. 5, McCormick SF (ed.)*. SIAM: Philadelphia, PA., 1986.
- [3] Brandt A, Diskin B. Multigrid solvers on decomposed domains. *Domain Decomposition Methods in Science and Engineering: The Sixth International Conference on Domain Decomposition, Contemporary Mathematics*, vol. 157, American Mathematical Society: Providence, Rhode Island, 1994; 135–155.
- [4] Mitchell W. A parallel multigrid method using the full domain partition. *Electron. Trans. Numer. Anal.* 1998; **6**:224–233.
- [5] Mitchell W. Parallel adaptive multilevel methods with full domain partitions. *App. Num. Anal. and Comp. Math.* 2004; **1**:36–48.
- [6] Bank RE, Holst MJ. A new paradigm for parallel adaptive meshing algorithms. *SIAM J. Sci. Stat. Comp.* 2000; **22**:1411–1443.
- [7] Bank RE, Lu S. A domain decomposition solver for a parallel adaptive meshing paradigm. *SIAM J. Sci. Comput.* 2004; **26**(1):105–127.
- [8] Bank RE, Vassilevski PS. Convergence analysis of a domain decomposition paradigm. *Comput. Visual Sci.* 2008; **11**:333–350.
- [9] Bank RE. Some variants of the Bank and Holst parallel adaptive meshing paradigm. *Comput. Vis. Sci.* Oct 2006; **9**(3):133–144.
- [10] Bank RE, Jimack P. A new parallel domain decomposition method for the adaptive finite element solution of elliptic partial differential equations. *Concurrency Computat.: Pract. Exper.* 2001; **13**:327–350.
- [11] Bank RE, Lu S, Tong C, Vassilevski PS. Scalable parallel algebraic multigrid solvers. *Technical Report UCRL-TR-210788*, Lawrence Livermore National Laboratory, Livermore, California 2004.
- [12] Chacon L. An optimal, parallel, fully implicit newton-krylov solver for three-dimensional visco-resistive magnetohydrodynamics. *Phys. Plasmas* 2008; **15**:56–103.
- [13] Chacon L, Knoll D, Finn J. An implicit, nonlinear reduced resistive mhd solver. *J. Comp. Phys.* 2002; **178**:15–36.
- [14] Chacon L, Knoll D. A 2d high-beta hall mhd implicit nonlinear solver. *J. Comp. Phys.* 2003; **188**:573–592.
- [15] Brown PN, Saad Y. Hybrid krylov methods for nonlinear systems of equation. *SIAM J. Sci. Stat. Comp.* 1990; **11**:450–481.
- [16] Brown PN, Saad Y. Convergence theory of nonlinear newton-krylov algorithms. *SIAM J. Opt.* 1994; **4**:297–330.
- [17] Cai XC, Gropp WD, Keyes DE, Melvin RG, Young DP. Parallel newton-krylov-schwarz algorithms for the transonic full potential equation. *SIAM J. Sci. Comp.* 1998; **19**:246–265.
- [18] Knoll D, Keyes D. Jacobian-free newton-krylov methods: a survey of approaches and applications. *J. Comp. Phys.* 2004; **193**:357–397.
- [19] Chen Y, Shen C, Wang J. Distributed transient stability simulation of power systems based on a jacobian-free newton-gmres method. *IEEE Trans. Power Systems* 2009; **24**:146–156.
- [20] Heroux MA, Bartlett RA, Howle VE, Hoekstra RJ, Hu JJ, Kolda TG, Lehoucq RB, Long KR, Pawlowski RP, Phipps ET, et al.. An overview of the trilinos project. *ACM Trans. Math. Softw.* 2005; **31**(3):397–423, doi:<http://doi.acm.org/10.1145/1089014.1089021>.
- [21] Nonlinear object-oriented solutions.
- [22] Pawlowski R, Shadid J, Simonis J, Walker H. Globalization techniques for newton-krylov methods and applications to the fully coupled solution of the navier-stokes equations. *SIAM Review* 2006; **48**(4):700–721, doi:[10.1137/S0036144504443511](https://doi.org/10.1137/S0036144504443511). URL <http://dx.doi.org/10.1137/S0036144504443511>.
- [23] AztecOO.
- [24] Gee M, Siefert C, Hu J, Tuminaro R, Sala M. ML 5.0 smoothed aggregation user's guide. *Technical Report SAND2006-2649*, Sandia National Laboratories 2006.
- [25] Loverich J, Zhou SCD, Beckwith K, Kundrapu M, Loh M, Mahalingam S, Stoltz P, Hakim A. *Nautlius: A Tool For Modeling Fluid Plasmas*. American Institute of Aeronautics and Astronautics, 2013, doi:[10.2514/6.2013-1185](https://doi.org/10.2514/6.2013-1185). URL <http://dx.doi.org/10.2514/6.2013-1185>.
- [26] Hawley JF, Stone JM. MOCCT: A numerical technique for astrophysical MHD. *Computer Physics Communications* Aug 1995; **89**:127–148, doi:[10.1016/0010-4655\(95\)00190-Q](https://doi.org/10.1016/0010-4655(95)00190-Q).
- [27] Dedner A, Kemm F, Kröner D, Munz CD, Schnitzer T, Wesenberg M. Hyperbolic Divergence

Cleaning for the MHD Equations. *Journal of Computational Physics* Jan 2002; **175**:645–673, doi:10.1006/jcph.2001.6961.

[28] Mignone A, Tzeferacos P. A second-order unsplit Godunov scheme for cell-centered MHD: The CTU-GLM scheme. *Journal of Computational Physics* Mar 2010; **229**:2117–2138, doi:10.1016/j.jcp.2009.11.026.

[29] Stone JM, Gardiner TA, Teuben P, Hawley JF, Simon JB. Athena: A New Code for Astrophysical MHD. *ApJSS* Sep 2008; **178**:137–177, doi:10.1086/588755. URL <http://adsabs.harvard.edu/abs/2008ApJS..178..137S>.

[30] Roe PL. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *Journal of Computational Physics* Oct 1981; **43**:357–372, doi:10.1016/0021-9991(81)90128-5.

[31] Nejat A, Ollivier-Gooch C. A high-order accurate unstructured finite volume Newton Krylov algorithm for inviscid compressible flows. *Journal of Computational Physics* Feb 2008; **227**:2582–2609, doi:10.1016/j.jcp.2007.11.011.

[32] Gardiner TA, Stone JM. An unsplit Godunov method for ideal MHD via constrained transport. *Journal of Computational Physics* May 2005; **205**:509–539, doi:10.1016/j.jcp.2004.11.016. URL <http://adsabs.harvard.edu/abs/2005JCoPh.205..509G>.

[33] Gardiner TA, Stone JM. An unsplit Godunov method for ideal MHD via constrained transport in three dimensions. *Journal of Computational Physics* Apr 2008; **227**:4123–4141, doi:10.1016/j.jcp.2007.12.017. URL <http://adsabs.harvard.edu/abs/2008JCoPh.227.4123G>.

[34] Jackson JD. *Classical electrodynamics*; 3rd ed. Wiley: New York, NY, 1999. URL <http://cds.cern.ch/record/490457>.

[35] Beckwith K, Stone JM. A Second-order Godunov Method for Multi-dimensional Relativistic Magnetohydrodynamics. *ApJSS* Mar 2011; **193**:6, doi:10.1088/0067-0049/193/1/6.

[36] Beckwith K, Stoltz P, Kundrapu M. Verification and validation tests of a two-fluid plasma model in mhd-like form 2016. In preparation.

[37] Chen G, Chacón L, Leibs C, Knoll D, Taitano W. Fluid preconditioning for newton-krylov-based, fully implicit, electrostatic particle-in-cell simulations. *Journal of Computational Physics* 2014; **258**(0):555–567.

[38] Kaufman AN, Rostler PS. The darwin model as a tool for electromagnetic plasma simulation. *Physics of Fluids* 1971; **14**:446–448.

[39] Krause TB, Apte A, Morrison P. A unified approach to the darwin approximation. *Physics of Fluids* 2007; **14**:102112.

[40] Nielson CW, Lewis HR. Particle-code models in the nonradiative limit. *Methods in Computational Physics* 1976; **16**:367–388.

[41] Arnold DN, Falk RS, Winther R. Preconditioning in  $h(\text{div})$  and applications. *Mathematics of Computation* 1997; **66**(219):957–984.

[42] Arnold DN, Falk RS, Winther R. Multigrid in  $h(\text{div})$  and  $h(\text{curl})$ . *Numerische Mathematik* 2000; **85**(2):197–217.

[43] Manteuffel TA, McCormick SF, Pflaum C. Improved discretization error estimates for first-order system least-squares (folsls). *J. of Numer. Math.* 2003; **11**:163–177.

[44] Falgout RD, Friedhoff S, Kolev TV, MacLachlan SP, Schroder JB. Parallel time integration with multigrid. *SIAM Journal on Scientific Computing* 2014; **36**(6):C635–C661, doi:10.1137/130944230.

[45] Falgout RD, Katz A, Kolev T, Schroder JB, Wissink A, Yang UM. Parallel time integration with multigrid reduction for a compressible fluid dynamics application. *Journal of Computational Physics* 2015; (**submitted**).

[46] Nievergelt J. Parallel methods for integrating ordinary differential equations. *Comm. ACM* 1964; **7**:731–733.

[47] Gander MJ. *50 years of Time Parallel Time Integration*. Multiple Shooting and Time Domain Decomposition, Springer, 2015. In press.

[48] Lions JL, Maday Y, Turinici G. Résolution d'EDP par un schéma en temps “pararéel”. *C. R. Acad. Sci. Paris Sér. I Math.* 2001; **332**(7):661–668.

[49] Gander MJ, Vandewalle S. Analysis of the parareal time-parallel time-integration method. *SIAM Journal on Scientific Computing* 2007; **29**:556–578.

[50] Calhoun J, Olson L, Snir M. FlipIt: An LLVM based fault injector for HPC. *Euro-Par 2014: Parallel Processing Workshops, Lecture Notes in Computer Science*, vol. 8805, Lopes L, Ziliinskas J, Costan A, Casella RG, Kecskemeti G, Jeannot E, Cannataro M, Ricci L, Benkner S, Petit S, et al. (eds.). Springer International Publishing, 2014; 547–558, doi:10.1007/978-3-319-14325-5\_47.

[51] Calhoun J, Snir M, Olson L, Garzaran M. Understanding the propagation of error due to a silent data corruption in a sparse matrix vector multiply. *IEEE Cluster 2015*, 2015. Best Poster Award.

- [52] Calhoun J, Olson LN, Snir M, Gropp WD. Towards a more fault resilient multigrid solver. *Proceedings of the High Performance Computing Symposium*, HPC 15, Society for Computer Simulation International: San Diego, CA, USA, 2015.
- [53] Beckwith K, Veitzer S, McCormick S, Ruge J, Olson L, Calhoun J. Fully-implicit ultrascale physics solvers and application to ion source modelling. *Plasma Sciences (ICOPS) held with 2014 IEEE International Conference on High-Power Particle Beams (BEAMS), 2014 IEEE 41st International Conference on*, 2014; 1–8, doi:10.1109/PLASMA.2014.7012726.
- [54] Calhoun J, Olson L, Snir M. Understanding error propagation in scientific codes 2016. In preparation.

1.

**1. Report Type**

Final Report

**Primary Contact E-mail**

**Contact email if there is a problem with the report.**

stevem@colorado.edu

**Primary Contact Phone Number**

**Contact phone number if there is a problem with the report**

970-944-2580

**Organization / Institution name**

Front Range Scientific Computations, Inc.

**Grant/Contract Title**

**The full title of the funded effort.**

Parallel Multilevel Decomposition Methods for Fluid Plasma Models

**Grant/Contract Number**

**AFOSR assigned control number. It must begin with "FA9550" or "F49620" or "FA2386".**

FA9550-12-1-0478

**Principal Investigator Name**

**The full name of the principal investigator on the grant or contract.**

Stephen F. McCormick

**Program Manager**

**The AFOSR Program Manager currently assigned to the award**

Jean-Luc Cambier

**Reporting Period Start Date**

09/29/2012

**Reporting Period End Date**

01/01/2016

**Abstract**

Progress of this project was on the following topics:

1. Multilevel Decomposition. Full implementations of both domain and range versions were developed and the comparative advantages and disadvantages were studied and documented. Numerical tests and model analyses showed these methods to be effective for current and foreseeable large-scale parallel architectures.

2. Adaptive AMG. New adaptive versions of AMG/SA were developed and tested, showing that the large-scale performance characteristics of these full-system methods to be superior to conventional approaches.

3. Jacobian-Free Newton-Krylov Methods. New physics-based preconditioners for the two-fluid problems were developed that exploit the new, ultra-scalable multilevel algorithms. New strategies for reducing preconditioner setup costs during the JFNK iterations were developed and implemented.

4. Space-Time Parallelization. A new multilevel algorithm was developed for implicit time-stepping schemes that coarsens in space and time and its efficacy for space-time grids was demonstrated.

DISTRIBUTION A: Distribution approved for public release

5. Fault Resilience. A software layer for fault detection and recovery was developed that seamlessly integrates into the multilevel decomposition methods. Efficiency of the proposed fault detection and recovery method was carefully demonstrated.

#### **Distribution Statement**

This is block 12 on the SF298 form.

Distribution A - Approved for Public Release

#### **Explanation for Distribution Statement**

If this is not approved for public release, please provide a short explanation. E.g., contains proprietary information.

#### **SF298 Form**

Please attach your [SF298](#) form. A blank SF298 can be found [here](#). Please do not password protect or secure the PDF

The maximum file size for an SF298 is 50MB.

[sf298.pdf](#)

**Upload the Report Document. File must be a PDF. Please do not password protect or secure the PDF . The maximum file size for the Report Document is 50MB.**

[AFFR-001.pdf](#)

**Upload a Report Document, if any. The maximum file size for the Report Document is 50MB.**

#### **Archival Publications (published) during reporting period:**

Beckwith K, Veitzer S, McCormick S, Ruge J, Olson L, Calhoun J. Fully-implicit ultra scale physics solvers and application to ion source modelling. Plasma Sciences (ICOPS) held with 2014 IEEE International Conference on High-Power Particle Beams (BEAMS), 2014 IEEE 41st International Conference on, 2014; 1{8,doi:10.1109/PLASMA.2014.7012726.

#### **Changes in research objectives (if any):**

#### **Change in AFOSR Program Manager, if any:**

Jean-Luc Cambier replaced Fariba Fahroo

#### **Extensions granted or milestones slipped, if any:**

Extension granted from September 29, 2015, to January 1, 2016.

#### **AFOSR LRIR Number**

#### **LRIR Title**

#### **Reporting Period**

#### **Laboratory Task Manager**

#### **Program Officer**

#### **Research Objectives**

#### **Technical Summary**

#### **Funding Summary by Cost Category (by FY, \$K)**

|                      | Starting FY | FY+1 | FY+2 |
|----------------------|-------------|------|------|
| Salary               |             |      |      |
| Equipment/Facilities |             |      |      |
| Supplies             |             |      |      |
| Total                |             |      |      |

#### **Report Document**

#### **Report Document - Text Analysis**

#### **Report Document - Text Analysis**

DISTRIBUTION A: Distribution approved for public release

**Appendix Documents**

**2. Thank You**

**E-mail user**

Feb 23, 2016 15:01:08 Success: Email Sent to: stevem@colorado.edu